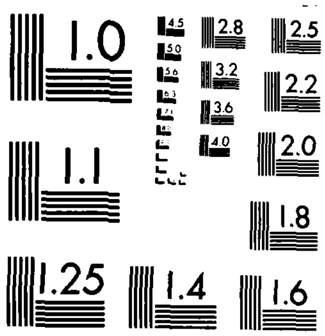
AD-A155 137 PARAMETER ESTIMATION IN COMMUNICATION SYSTEM TRACKING SATELLITE OBSERVATIONS(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA V A TSAFARAS DEC 84 1/1

UNCLASSIFIED F/G 17/2 NL

END



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(2)

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

PARAMETER ESTIMATION IN COMMUNICATION
SYSTEM TRACKING SATELLITE OBSERVATIONS

by

Vassilios Ath. Tsafaras

December 1984

Thesis Advisor: H. A. Titus

Approved for public release; distribution unlimited

DISCLAIMER NOTICE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A155187	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Parameter Estimation in Communication System Tracking Satellite Observations		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; December 1984
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Vassilios Ath. Tsafaras		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		12. REPORT DATE December 1984
		13. NUMBER OF PAGES 82
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Parameter Estimation; Kalman Filter,

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The estimation of parameters from a satellite communication system is often through the use of Kalman filtering. In this work the location of the eye of a hurricane is estimated from satellite observations. A comparison with a posteriori meteorologist's analysis was attempted. An adaptive gating scheme was employed in the filter to accommodate "manueuvers" of the storm.

The observations were at random intervals and also came from several different sources (aircraft and land based radar as well as satellite).

Approved for public release; distribution unlimited

Parameter Estimation in Communication System Tracking Satellite Observations

by

Vassilios Ath. Tsafaras
Lieutenant, Hellenic Navy
B.S., Greek Naval Academy, 1973

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1984

Author: Vassilios Ath. Tsafaras

Approved by: H.A. Titus, Thesis Advisor

A. Gerba, Jr., Second Reader

H. Rigas, Chairman, Department of Electrical and Computer Engineering

J. Dyer, Dean of Science and Engineering

ABSTRACT

The estimation of parameters from a satellite communication system is often through the use of Kalman filtering. In this work the location of the eye of a hurricane is estimated from satellite observations. A comparison with a posteriori meteorologist's analysis was attempted. An adaptive gating scheme was employed in the filter to accommodate "manueuvers" of the storm.

The observations were at random intervals and also came from several different sources (aircraft and land based radar as well as satellite).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

Satellite communications, especially digital, are well under way. One of the many observations that satellites provide are the meteorological. These give the location of a storm or a typhoon in terms of geological coordinates. It is possible to have images of the store or typhoon cloud cover.

These observations occur quite randomly in time. Meteorologists like to present their forecasts equally spaced in time.

The attempt, here, is to try to compare an adaptive Kalman filtering algorithm to estimate the location of the eye of a storm with optimum track values that are given from the meteorologist's analysis. The filtering process combines all the available measurement data with prior knowledge about the system. It produces an estimate of the location of the storm (latitude, longitude) in such a manner that the mean square error is minimized. The parameters, which are essential design elements of a Kalman filter, are the measurement noise covariance matrix, R, the excitation covariance, Q, the initialization covariance of error in the filter itself, P(1/0), and the transition matrix, ϕ.

In most physical processes that one desired to track, many of these parameters change during the process of

tracking. The measurement noise associated with the observations can change if a different sensor is used, or a similar sensor obtaining measurements from some different geometry relative to the object being tracked. If the object being observed is acted upon by external forces, then the Q matrix should be changed to account for these external excitations. Most processes being tracked will change in their dynamic characteristics during the observation time and so the transition matrix ideally should also be changed. Further, the time between observations quite often occurs randomly in time. All of these things bring about the need to change the Kalman filter to adapt as the process changes.

It is possible to change the parameters of the Kalman filter by sensing the error between the observation and the prediction from the track of what that observation should be. If a gate is established representing 95% of the normal random perturbations of the process, then when this error exceeds the magnitude of the gate, one can reasonably ascertain that the filter is no longer properly representing the observed process. In the work attacked here, real data was obtained from satellite observations and the qualitative observation error was established (PCN #).

The error covariance matrix in terms of error ellipsoids along the track gives a measurement for the worthiness of the algorithm parameters.

II. KALMAN FILTERING TECHNIQUES

In a linear, discrete system, the state and measurement equations are given by

$$\underline{x}(k+1) = \underline{\phi}(k)\underline{x}(k) + \underline{\Gamma}w(k)$$

and

$$\underline{z}(k) = \underline{H}(k)\underline{x}(k) + \underline{v}(k)$$

where $\underline{x}$ is the state; $\underline{\phi}$ is the transition matrix; $\underline{\Gamma}$ is excitation noise matrix; $\underline{H}$ is the measurement matrix; $\underline{w}$ and $\underline{v}$ are the excitation and measurement noise correspondingly, assumed uncorrelated, zero mean white Gaussian:

$$E[\underline{w}(k)\cdot\underline{w}^T(j)] = Q(k)\delta_{kj}$$

and

$$E[\underline{v}(k)\cdot\underline{v}^T(j)] = R(k)\delta_{kj}$$

and

$$E[\underline{w}(k)] = 0, \; E[\underline{v}(k)] = 0$$

where Q(k) and R(k) are covariances of excitation and measurement noise. Now if $\underline{\hat{x}}(k)$ is the estimated state value after the kth measurement and $\hat{x}(k|k-1)$ is the predicted value of the state before the kth measurement we have:

$$\hat{\underline{x}}(k|k-1) = \underline{f}(\hat{\underline{x}}(k-1|k-1), k-1), \text{ where f is any function.}$$

The state error vector is defined to be

$$\tilde{\underline{x}}(k) = \hat{\underline{x}}(k) - \underline{x}(k)$$

and the predicted state error vector is defined to be

$$\tilde{\underline{x}}(k|k-1) = \hat{\underline{x}}(k|k-1) - \underline{x}(k).$$

The covariance of state error matrix is defined to be

$$\underline{P}(k|k) = E[\tilde{\underline{x}}(k) \cdot \tilde{\underline{x}}^T(k)],$$

and the predicted covariance of state error is defined as

$$\underline{P}(k|k-1) = E[\tilde{\underline{x}}(k|k-1) \cdot \tilde{\underline{x}}^T(k|k-1)].$$

The state excitation matrix is defined by

$$\underline{Q}(k) = \underline{\Gamma}(k) E[\underline{w}(k) \cdot \underline{w}^T(k)] \cdot \underline{\Gamma}^T(k),$$

and the measurement noise covariance matrix is defined by

$$\underline{R}(k) = E[\underline{v}(k) \cdot \underline{v}^T(k)].$$

The Kalman filter equations are:

$$\underline{P}(\text{[illegible]}) = \underline{\Phi}(k) \underline{P}(k|k) \underline{\Phi}^T(k) + \underline{Q}(k)$$

$$\underline{G}(k) = \underline{P}(\text{[illegible]}k-1) \underline{H}^T(k) [\underline{H}(k) \underline{P}(k|k-1) \underline{H}^T(k) + \underline{R}(k)]^{-1}$$

$$\underline{P}(k|k) = [\text{[illegible]} - \underline{G}(k) \underline{H}(k)] \underline{P}(k|k-1)$$

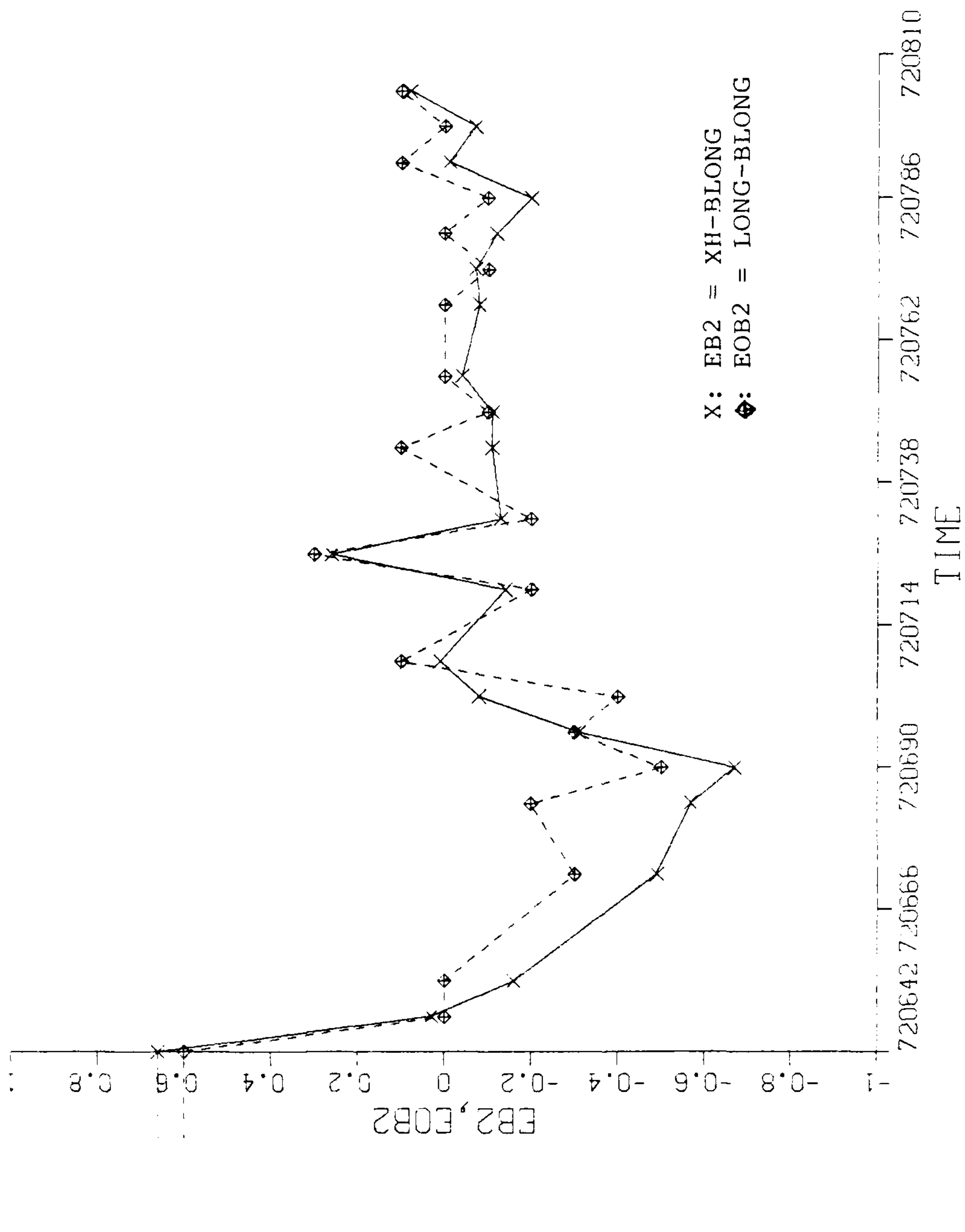


Figure 5 Longitude Errors

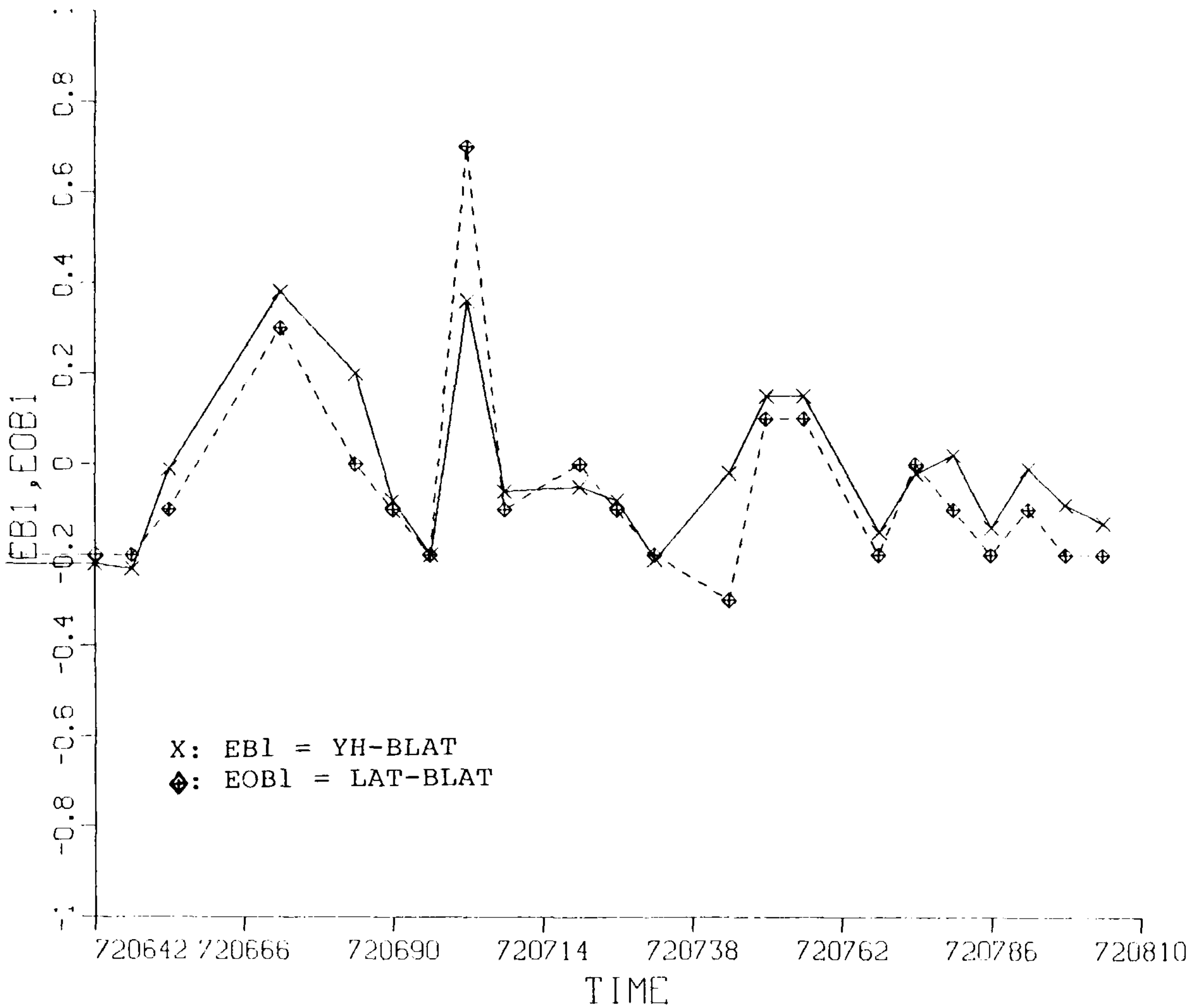


Figure 4 Latitude Errors

TABLE 3

ERROR VALUES IN TERMS OF LATITUDE AND LONGITUDE

ITIME	EB1	EB2	EOB1	EOB2
720642	-0.22	0.06	-0.20	0.60
720648	-0.23	0.03	-0.20	0.0
720654	-0.01	-0.16	-0.10	0.0
720672	0.38	-0.44	0.30	-0.30
720684	0.20	-0.37	0.0	-0.20
720690	-0.08	-0.67	-0.10	-0.50
720696	-0.20	-0.31	-0.20	-0.30
720702	0.36	-0.08	0.70	-0.40
720708	-0.06	0.01	-0.10	0.10
720720	-0.05	-0.14	0.0	-0.20
720726	-0.08	0.26	-0.10	0.30
720732	-0.21	-0.13	-0.20	-0.20
720744	-0.02	-0.11	-0.30	0.10
720750	0.15	-0.11	0.10	-0.10
720756	0.15	-0.04	0.10	0.0
720768	-0.15	-0.08	-0.20	0.0
720774	-0.02	-0.07	0.0	-0.10
720780	0.02	-0.12	-0.10	0.0
720786	-0.14	-0.20	-0.20	-0.10
720792	-0.01	-0.01	-0.10	0.10
720798	-0.09	-0.07	-0.20	0.0
720804	-0.13	0.08	-0.20	0.10
720810	-0.04	0.05	0.0	0.0
720816	-0.08	-0.17	0.0	-0.30

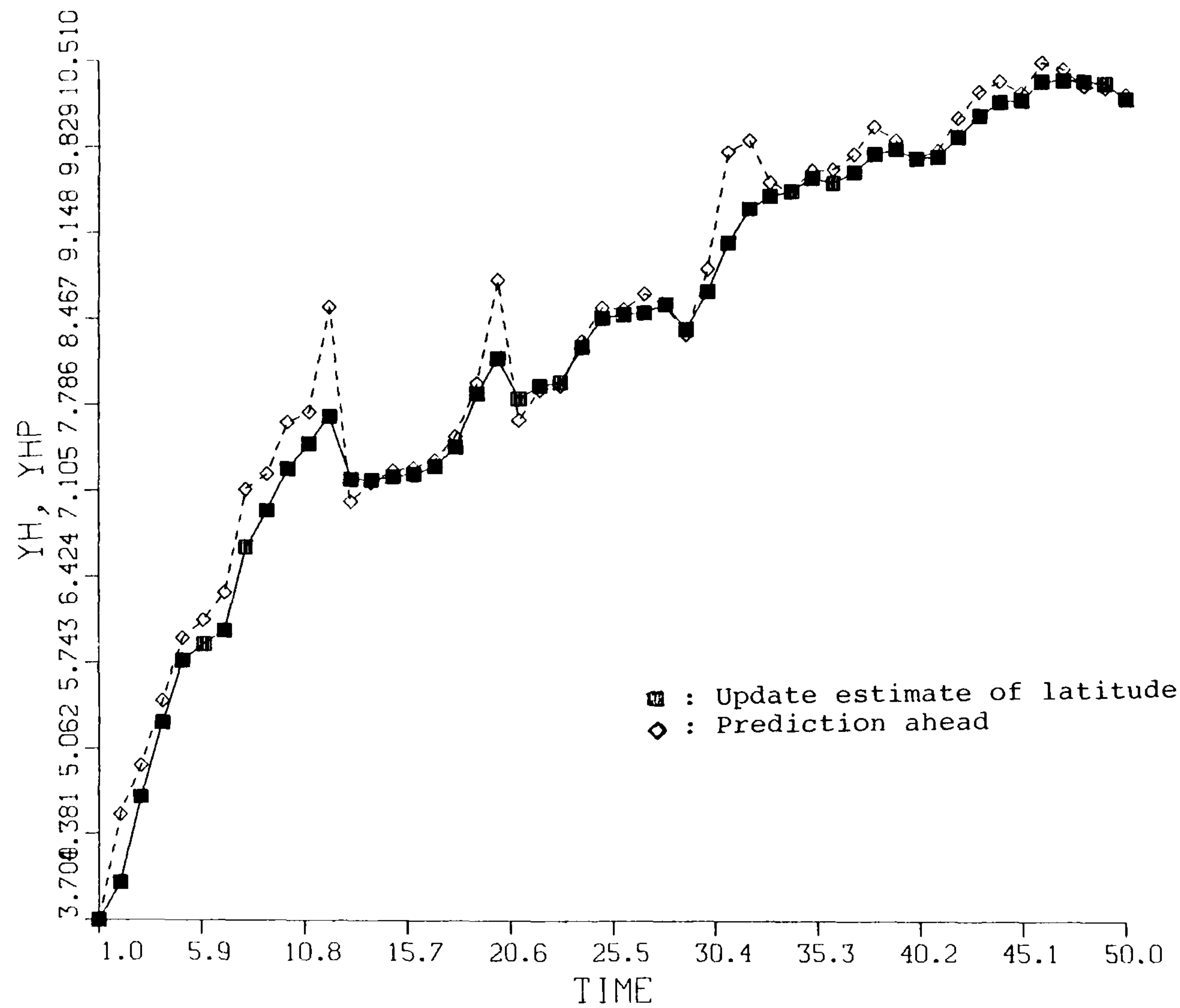


Figure 3 K.F Track and Prediction Ahead in Latitude

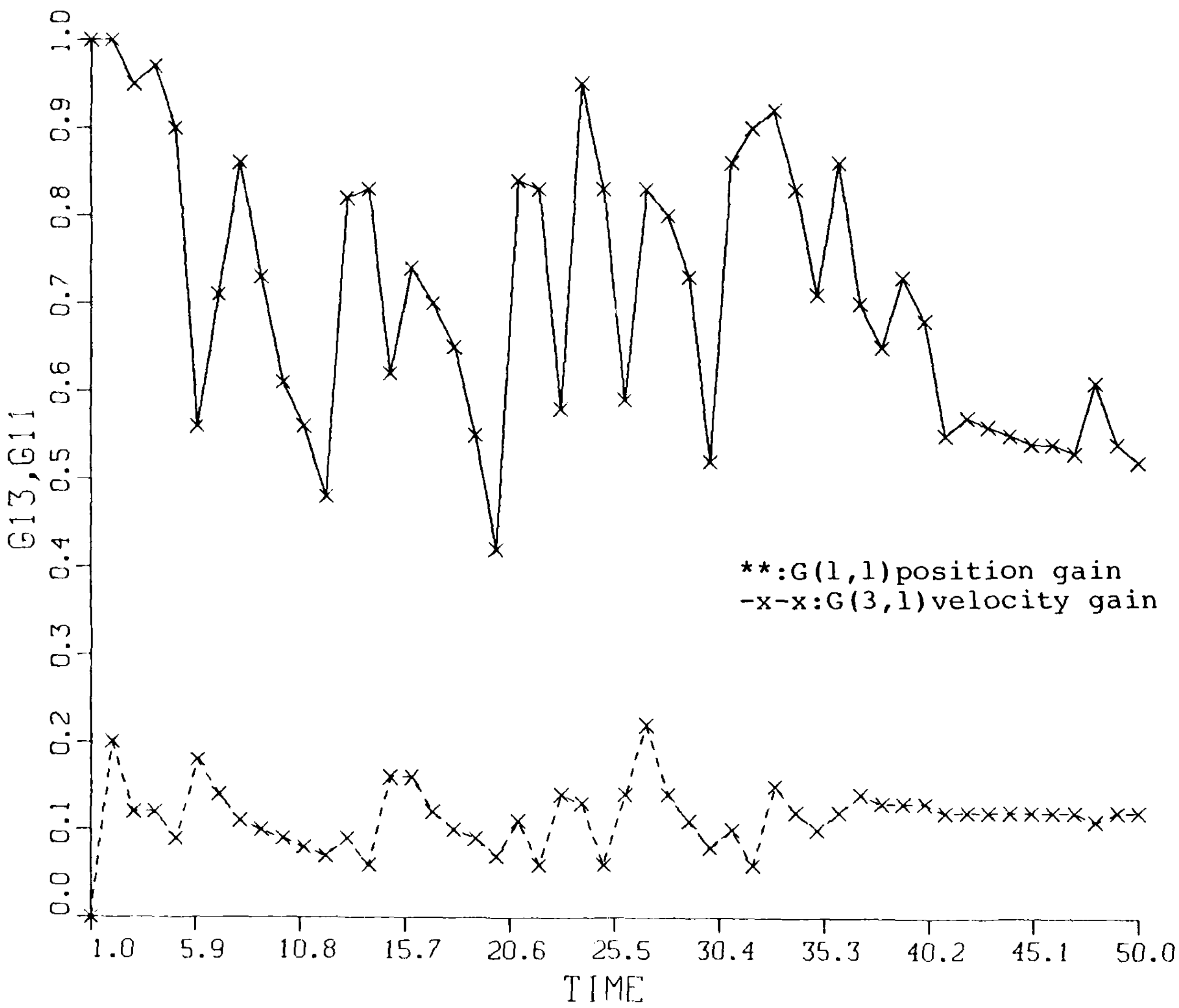


Figure 2 Gains of Position and Velocity

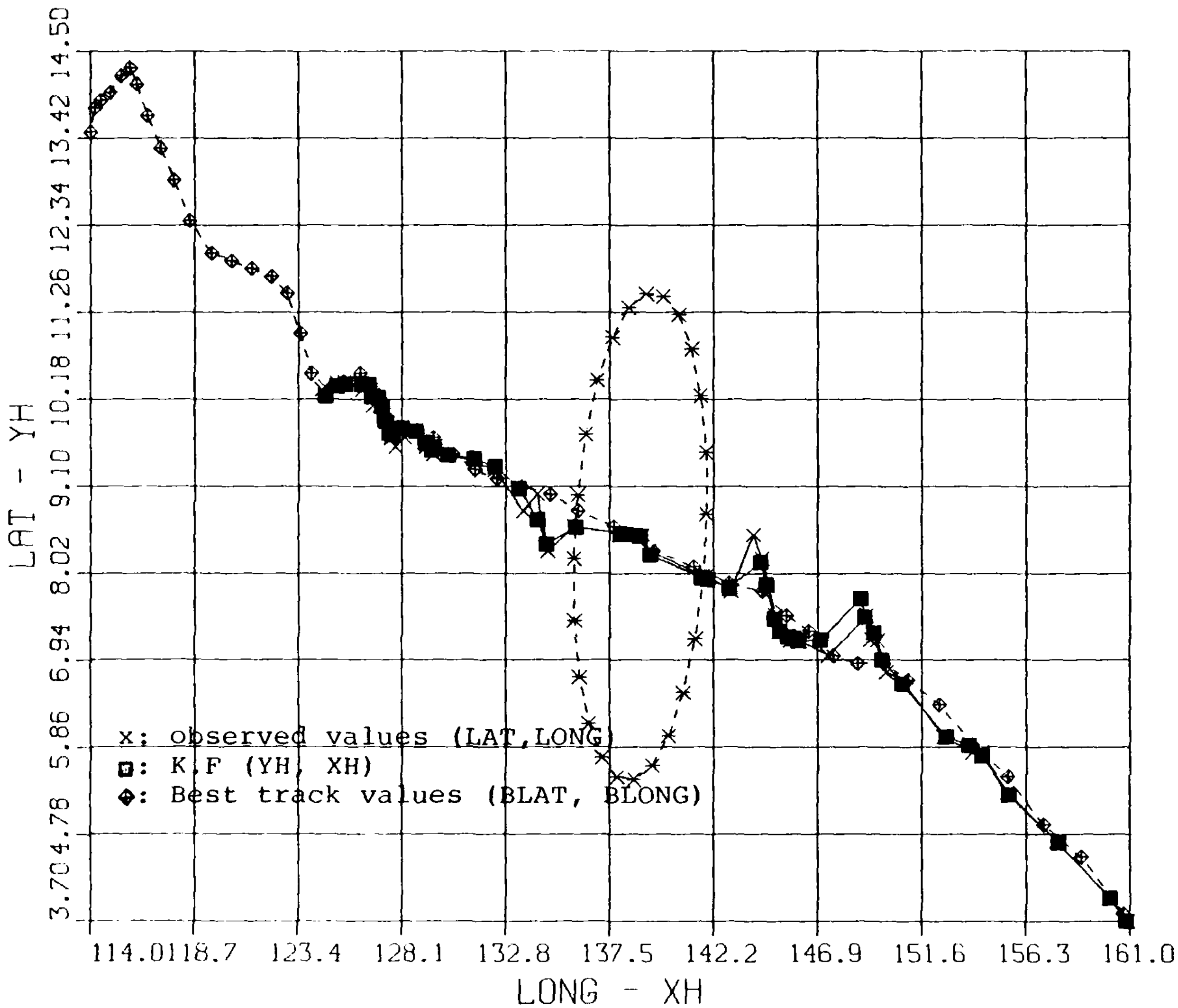


Figure 1 Trajectory of Storm Nelson

TABLE 2

BEST TRACK DATA

TIME	BLAT	BLONG
1806.00	3.80	160.70
1812.00	4.50	158.80
1818.00	4.90	157.10
1900.00	5.50	155.50
1906.00	5.90	153.90
1912.00	6.40	152.40
1918.00	6.70	151.00
2000.00	6.90	149.90
2006.00	6.90	148.70
2012.00	7.00	147.60
2018.00	7.30	146.50
2100.00	7.50	145.50
2106.00	7.80	144.40
2112.00	7.90	142.90
2118.00	8.10	141.30
2200.00	8.30	139.50
2206.00	8.60	137.70
2212.00	8.80	136.10
2218.00	9.00	134.80
2300.00	9.10	133.50
2306.00	9.20	132.40
2312.00	9.30	131.40
2318.00	9.50	130.40
2400.00	9.70	129.50
2406.00	9.80	128.80
2412.00	9.80	128.20
2413.00	9.90	127.70
2500.00	10.10	127.20
2506.00	10.30	126.80
2512.00	10.50	126.20
2518.00	10.40	125.50
2600.00	10.30	124.80
2606.00	10.50	124.00
2612.00	11.00	123.50
2618.00	11.50	122.90
2700.00	11.70	122.20
2706.00	11.80	121.30
2712.00	11.90	120.40
2718.00	12.00	119.50
2800.00	12.40	118.50
2806.00	12.90	117.80
2812.00	13.30	117.20
2818.00	13.70	116.60
2900.00	14.10	116.10
2906.00	14.30	115.80
2912.00	14.20	115.40
2918.00	14.00	114.90
3000.00	13.90	114.50
3006.00	13.80	114.20
3012.00	13.50	114.00

TABLE 1

OBSERVED DATA (SATELLITE FIXES)

TIME	LAT	LONG
1804.00	3.70	160.90
1809.00	4.00	160.10
1818.00	4.70	157.70
1900.00	5.30	155.50
1903.48	5.80	154.30
1906.00	5.80	153.90
1909.00	6.00	152.60
1916.33	6.70	150.70
1921.00	6.80	150.00
2000.00	7.20	149.60
2003.00	7.20	149.30
2005.18	7.50	149.10
2012.00	7.00	147.40
2016.21	7.20	146.10
2018.00	7.20	146.00
2021.00	7.20	145.70
2100.00	7.30	145.20
2103.00	7.50	145.00
2105.06	8.20	144.40
2106.00	8.50	144.00
2112.00	7.80	143.00
2116.00	8.00	141.90
2117.51	8.00	141.70
2200.00	8.30	139.30
2203.00	8.50	139.00
2204.54	8.50	138.10
2206.00	8.50	138.00
2212.00	8.60	135.90
2216.00	8.30	134.70
2217.40	9.00	134.20
2300.00	8.80	133.60
2306.00	9.30	132.30
2312.00	9.40	131.40
2317.28	9.50	130.10
2321.00	9.60	129.60
2400.00	9.50	129.50
2403.00	9.60	129.20
2406.13	9.80	128.70
2412.00	9.70	128.20
2416.00	9.60	127.80
2418.00	9.70	127.60
2421.00	9.90	127.40
2500.00	10.00	127.30
2503.00	10.10	127.10
2506.01	10.10	126.80
2509.00	10.30	126.70
2512.00	10.30	126.30
2518.00	10.40	125.50
2521.00	10.40	125.10
2600.00	10.30	124.50

magnitude of the gate ($\sqrt{\underline{P}(k\ k-1) + \underline{R}}$) resulted in changing the values of the gains. The above correction makes the filter more adaptive now and the errors EB1 and EB2 appear smaller on an average. The representation of the above correction in terms of trajectories, gains, predictions and errors appear in Figures 6, 7, 8, 9, and 10 and Table 4. The computer program appears in Appendix A.

$$X(1/0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

In Figure 1 we have a representation of the best track observed and the K.F track. An error ellipsoid at the 25th stage of the process appears. It seems that K.F follows the observed track more closely than the best track. This is due to the adaptive gating and Q relative to R.

The gain history for G(1,1) fluctuates between 0.7-0.8, never arriving at a stable value. The G(3,1) reaches a stable value of 0.12.

The above appears in Figure 2. The track for prediction appears in Figure 3.

The errors EB1 and EOB1 represent (YH-BLAT) and (LAT-BLAT) respectively. They, along with EB2 and EOB2 appear in Table 3 and Figures 4 and 5.

Implementing Julian Time, we have a comparison in common time for 24 points only. It can be seen that in terms of the latitude error the K.F is close to the best track values.

In an attempt for better performance the $\underline{Q}$ matrix was changed in the algorithm. Also the innovation errors, in terms of latitude and longitude, exceeding the

IV. SATELLITE TRACKING-SIMULATION RESULTS

Data from the Annual Tropical Cyclone Data for the Typhoon-Nelson appear in Tables 1 and 2 in terms of longitude and latitude. The best track data appears in six-hour intervals for twelve days and the satellite fixes (observed) in random time intervals.

The data in the K.F algorithm parameters are:

$$\phi = \begin{bmatrix} 1 & 0 & DT & 0 \\ 0 & 1 & 0 & DT \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} DT^2/2 & 0 \\ 0 & DT^2/2 \\ DT & 0 \\ 0 & DT \end{bmatrix}$$

$$P(1/0) = \begin{bmatrix} 10^3 & 0 & 0 & 0 \\ 0 & 10^3 & 0 & 0 \\ 0 & 0 & 10^3 & 0 \\ 0 & 0 & 0 & 10^3 \end{bmatrix}$$

however, does not have its major and minor axis aligned with the coordinate system. Instead its axis (x', y'), comes from the following transformation:

$$x' = x\cos\theta + y\sin\theta$$

$$y' = y\cos\theta - x\sin\theta$$

where x, y are the latitude and longitude in our case with

$$\theta = 1/2 \tan^{-1}\left[\frac{2\mathrm{cov}(x,y)}{\sigma_x^2 - \sigma_y^2}\right]$$

where $\cos(x,y) = P_{12}(k\ k-1)$

$$\sigma_x^2 = P_{11}(k|k-1) \text{ and } \sigma_y^2 = P_{22}(k|k-1)$$

The new variances are calculated by

$$\sigma'^2_x = \frac{\sigma_x^2 + \sigma_y^2}{2} + \frac{\mathrm{cov}(xy)}{\sin 2\theta}$$

$$\sigma'^2_y = \frac{\sigma_x^2 + \sigma_y^2}{2} - \frac{\mathrm{cov}(xy)}{\sin 2\theta}$$

Incorporating the above equations, error ellipses are presented in subsequent figures with the satellite tracks.

III. ERROR ELLIPSOIDS

The error covariance matrix in each stage of a Kalman filter process gives insight into the quality of the track occuring.

The diagonal terms (P_{11} and P_{22}) are the variances of uncertainty in our knowledge of latitude and longitude. Their respective off diagonal terms are covariance between latitude and longitude.

The square roots of the diagonal terms gives us the rms errors in out estimates of longitude and latitude. Having the definition of the structure we are dealing with (in our case the satellite observations) and its uncertainties (expressed by the PCN number-actually by the values of $\underline{R}$) we can see how the K.F performs through its error covariance matrix. Expressing the P matrix in an ellipsoid of constant probability, one obtains a visual appreciation for the worthiness of the algorithm parameters. The representation requires that the errors are normally distributed.

The joint probability density function is:

$$e^{-1/2e^T(k|k-1)\underline{P}^{-1}(k|k-1)e(k|k-1)}$$

where $e(k|k-1)$ is the predicted state error vector. Setting the exponent equal to a constant value, we are going to have a curve which is an ellipse. This ellipse,

$$\hat{\underline{x}}(k|k-1) = \underline{\Phi}(k)\hat{\underline{x}}(k-1|k-1)$$

$$\hat{\underline{z}}(k|k-1) = \underline{H}(k)\hat{\underline{x}}(k|k-1)$$

$$\hat{x}(k|k) = \hat{\underline{x}}(k|k-1) + \underline{G}(k)[\underline{z}(k)-\hat{\underline{z}}(k|k-1)]$$

The initial condition of $\underline{P}$ (error covariance matrix) and the $\underline{Q}$ and $\underline{R}$ matrices are the determining factors in the filter structure. For $\underline{Q}$ having main diagonal values greater than $\underline{R}$ means that we have greater uncertainty in the state estimate than in the observation. Thus the new state estimate is more dependent upon the new measurement and less related to prior estimates. The inverse is also true. For $\underline{R}$ having greater diagonal terms indicates that the new measurement are subjected to stronger corruptive noises, and so should be weighted less by the filter. The gains (G) are lower. The $\underline{P}$ is responsible for the initial transient performance of the filter.

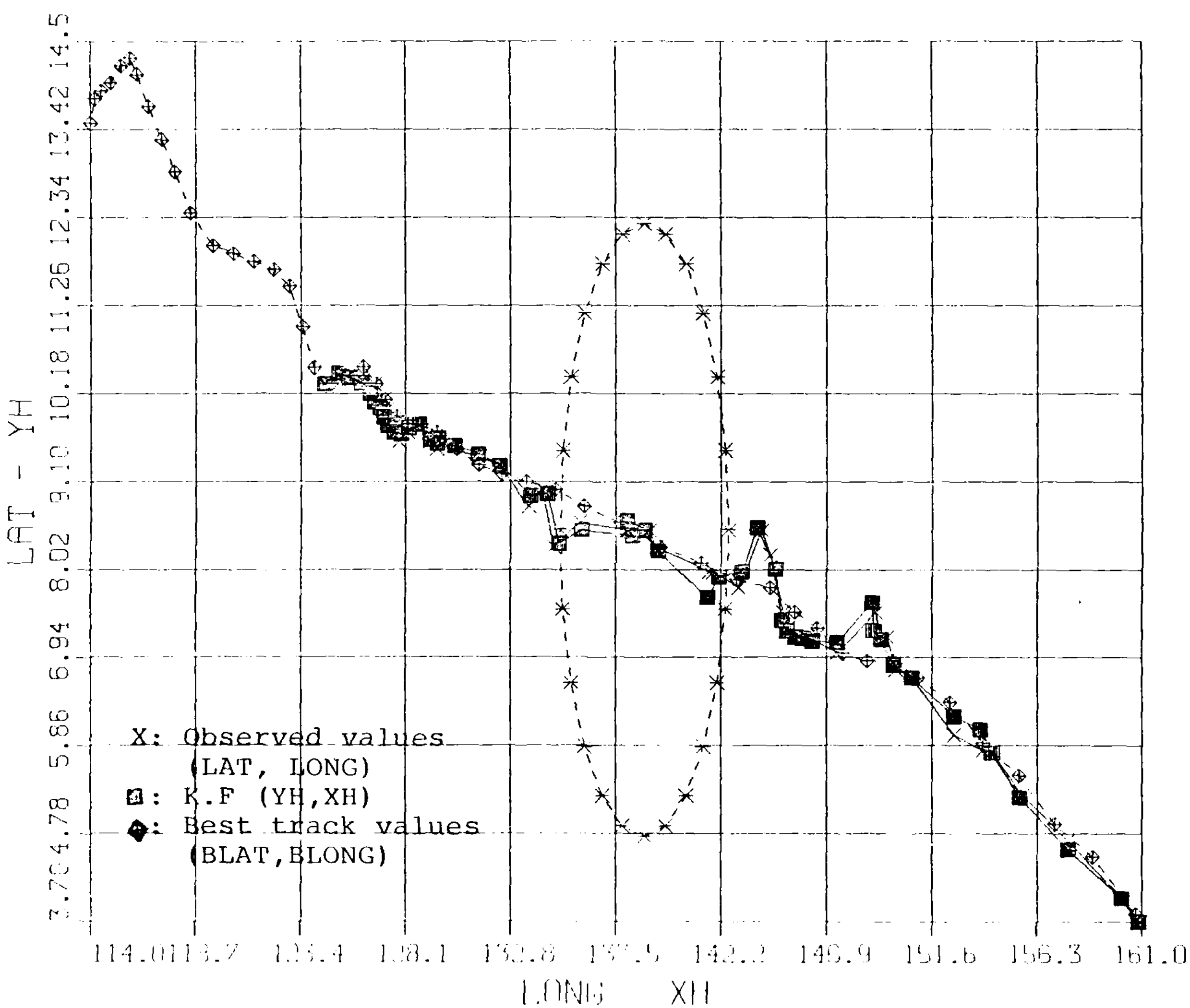


Figure 6 Trajectory of Storm Nelson

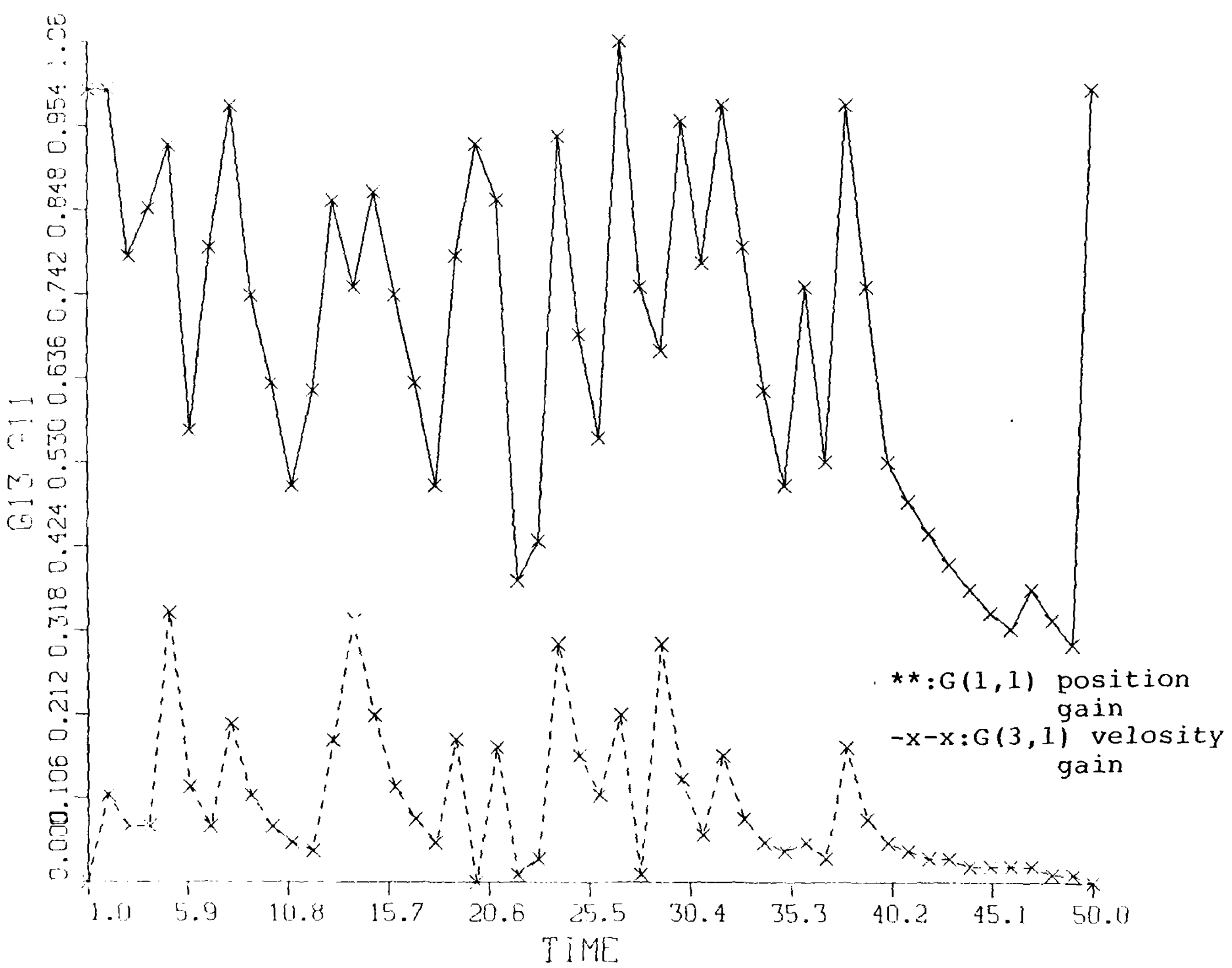


Figure 7 Gains of Position and Velocity

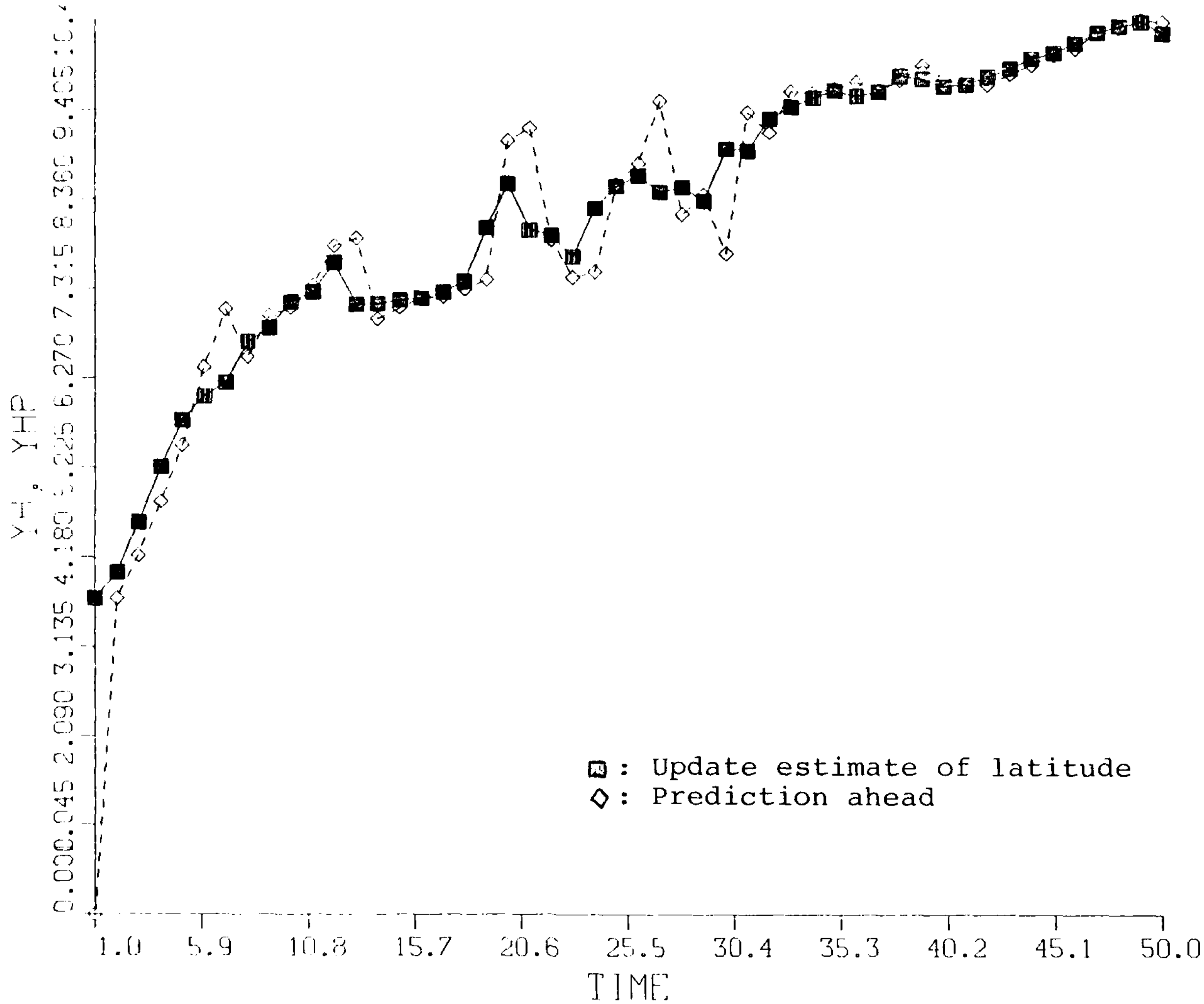


Figure 8 K.F Track and Prediction Ahead in Latitude

TABLE 4

ERROR VALUES IN TERMS OF LATITUDE AND LONGITUDE

JTIME	ES1	ES2	EC1	EC2
720642	-0.31	0.01	-0.20	0.60
720648	-0.27	0.03	-0.20	0.0
720654	0.15	-0.12	-0.10	0.0
720672	0.26	-0.57	0.30	-0.30
720684	0.13	-0.21	0.0	-0.20
720690	-0.12	-0.60	-0.10	-0.50
720696	-0.23	-0.75	-0.20	-0.30
720702	0.74	-0.77	0.70	-0.40
720708	0.10	0.13	-0.10	0.10
720720	-0.05	-0.12	0.0	-0.20
720726	-0.15	0.53	-0.10	0.30
720732	-0.30	-0.13	-0.20	-0.20
720744	-0.17	0.19	-0.50	0.10
720750	0.10	-0.05	0.10	-0.10
720756	0.14	-0.04	0.10	0.0
720768	-0.14	-0.01	-0.20	0.0
720774	-0.00	-0.10	0.0	-0.10
720780	-0.04	0.04	-0.10	0.0
720786	-0.20	-0.11	-0.20	-0.10
720792	-0.21	-0.07	-0.10	0.10
720798	-0.23	-0.09	-0.20	0.0
720804	-0.10	-0.06	-0.20	0.10
720810	-0.02	0.09	0.0	0.0
720816	-0.00	-0.30	0.0	-0.30

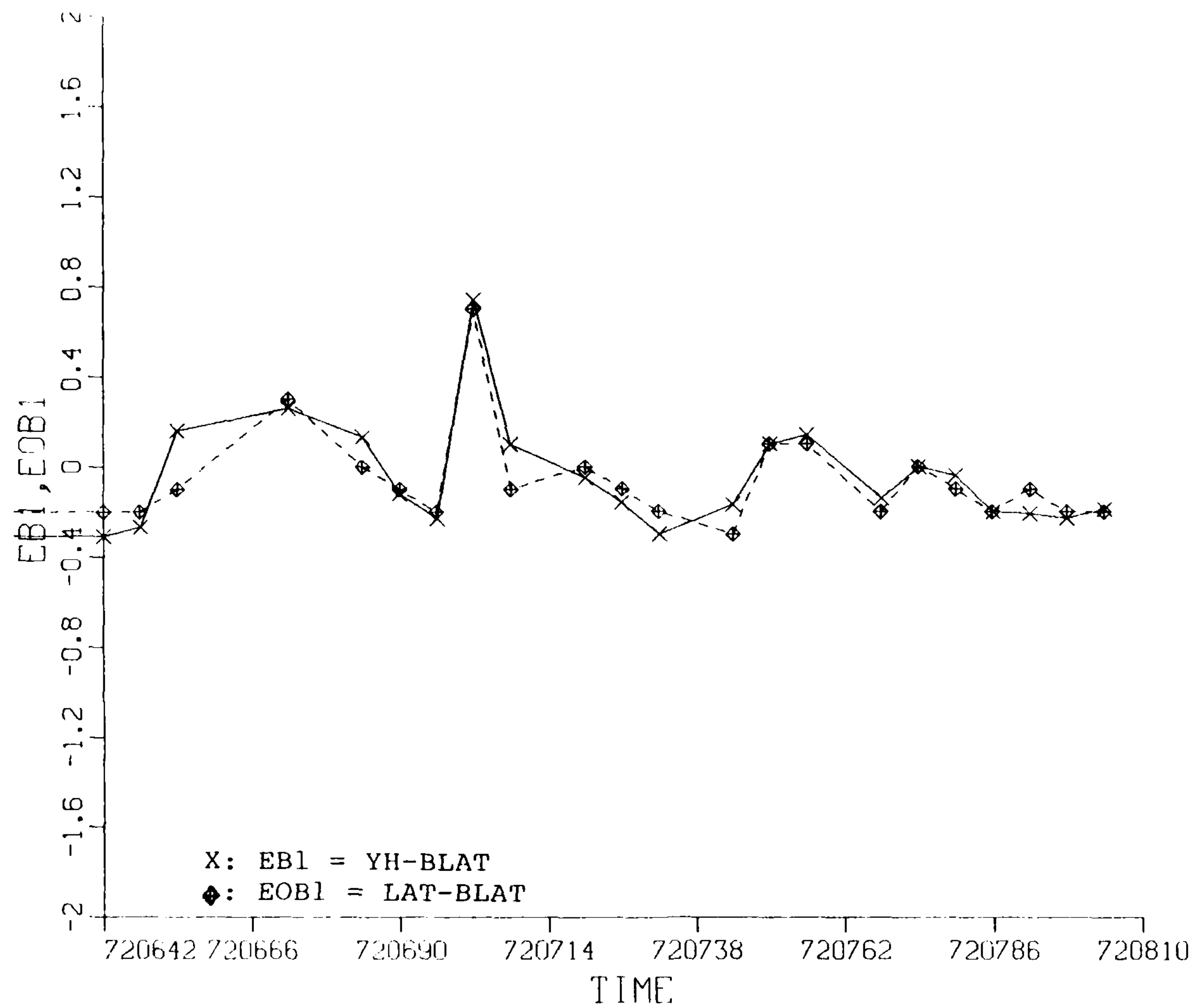


Figure 9 Latitude Errors

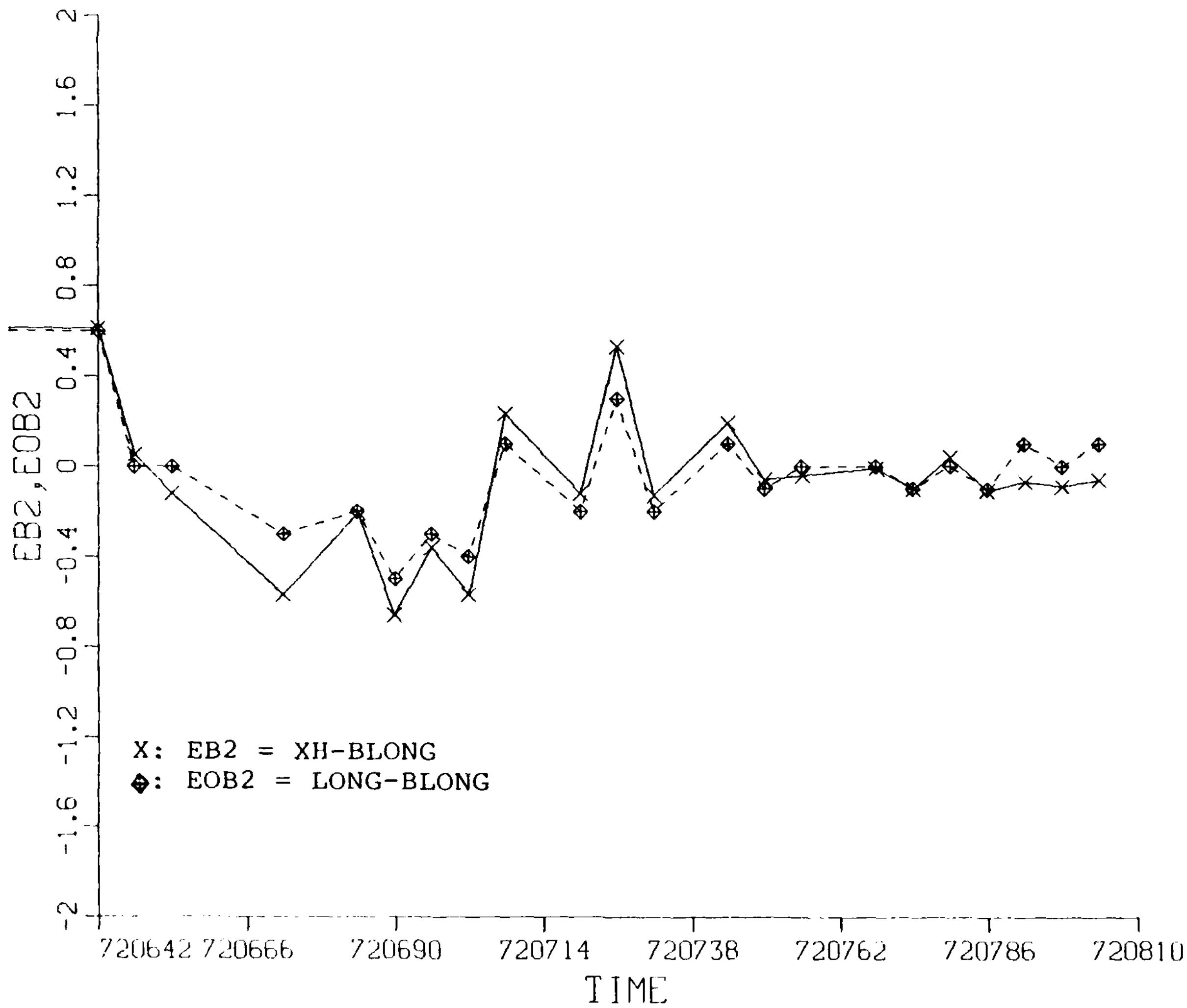


Figure 10 Longitude Errors

V. RANDOM TRACKING

To examine the adaptability of the K.F, another storm was created. The equations for this simulated storm were:

$$BLAT(s) = BLAT(s-1) - V_x \cdot T$$

$$BLONG(s) = BLONG(s-1) - V_y \cdot T$$

$$LAT(s) = BLAT(s) + V(s)$$

$$LONG(s) = BLONG(s) + V(s)$$

where T=6hr, V_x=10°/24hr, V_y=5°/24hr and V = measurement noise (created by a random generator subroutine).

Implementing the above equations, a "true" and an "observed" track were created. These two data files were named BESTRACK and OBSERVED and appear in Table 5 and 6.

Simulating with the above new data, the K.F algorithm appeared to track well. *Figure 11 shows the "true",* "observed" and K.F tracks.

Figure 12 indicates the gain history in terms of G11 and G13. This approach gave stable values of 0.27 and 0.01 respectively after the 4th discrete point in time. After this time the gain does not vary any more. This means that the innovation error, $(z(k)-\hat{x}(k|k-1))$ is

weighted each time by the same quantity after the 4th observation. Having lower values in the diagronal terms of the Q matrix in comparison with R, in this case, means that we have greater uncertainty in the measurements (observed data) relative to the model uncertainties. So the gains are smaller and the filter no longer "tracks" the measurements closely.

As far as the latitude and longitude errors are concerned it can be seen that EB1 (YH-BLAT) and EB2(XH-BLONG) are nearly zero and are smaller in comparison with EOB1 (LAT-BLAT), EOB2 (LONG-BLONG). This shows the ability of the algorithm to follow the "true" values more than the measurement if the latter has been corrupted with noise.

The error values and the plots appear in Table 7 and Figures 13 and 14.

TABLE 5

BESTRACK DATA

TIME	BLAT	BLONG
1800.00	160.00	3.80
1806.00	157.50	2.80
1812.00	155.00	1.80
1818.00	152.50	0.80
1900.00	150.00	-0.20
1906.00	147.50	-1.20
1912.00	145.00	-2.20
1918.00	142.50	-3.20
2000.00	140.00	-4.20
2006.00	137.50	-5.20
2012.00	135.00	-6.20
2018.00	132.50	-7.20
2100.00	130.00	-8.20
2106.00	127.50	-9.20
2112.00	125.00	-10.20
2118.00	122.50	-11.20
2200.00	120.00	-12.20
2206.00	117.50	-13.20
2212.00	115.00	-14.20
2218.00	112.50	-15.20
2300.00	110.00	-16.20
2306.00	107.50	-17.20
2312.00	105.00	-18.20
2318.00	102.50	-19.20
2400.00	100.00	-20.20
2406.00	97.50	-21.20
2412.00	95.00	-22.20
2418.00	92.50	-23.20
2500.00	90.00	-24.20
2506.00	87.50	-25.20
2312.00	85.00	-25.20
2518.00	82.50	-27.20
2600.00	80.00	-29.20
2606.00	77.50	-29.20
2512.00	75.00	-30.20
2618.00	72.50	-31.20
2700.00	70.00	-32.20
2706.00	67.50	-33.20
2712.00	65.00	-34.20
2718.00	62.50	-35.20
2800.00	60.00	-36.20
2806.00	57.50	-37.20
2312.00	55.00	-38.20
2818.00	52.50	-39.20
2900.00	50.00	-40.20
2906.00	47.50	-41.20
2712.00	45.00	-42.20
2918.00	42.50	-43.20
3000.00	40.00	-44.20
3006.00	37.50	-45.20

TABLE 6

OBSERVED DATA

TIME	LAT	LONG
1800.00	160.00	3.80
1806.00	157.30	2.60
1812.00	154.92	1.72
1818.00	152.71	1.01
1900.00	150.03	-0.17
1906.00	147.63	-1.07
1912.00	144.95	-2.25
1918.00	142.61	-3.09
2000.00	139.99	-4.21
2006.00	137.26	-5.44
2012.00	134.62	-6.58
2018.00	132.39	-7.31
2100.00	130.31	-7.89
2106.00	127.37	-9.33
2112.00	125.19	-10.01
2118.00	122.44	-11.26
2200.00	120.19	-12.01
2206.00	117.40	-13.30
2212.00	114.87	-14.33
2218.00	112.92	-14.78
2300.00	110.03	-16.17
2306.00	107.65	-17.05
2312.00	104.83	-18.37
2318.00	102.96	-18.74
2400.00	99.90	-20.30
2406.00	97.65	-21.05
2412.00	95.29	-21.91
2418.00	92.44	-23.26
2500.00	90.51	-23.69
2506.00	87.18	-25.52
2512.00	84.84	-26.36
2518.00	82.40	-27.30
2600.00	80.00	-28.20
2606.00	77.78	-28.92
2612.00	75.08	-30.12
2618.00	72.50	-31.20
2700.00	69.96	-32.24
2706.00	67.22	-33.48
2712.00	64.79	-34.41
2718.00	62.75	-34.95
2800.00	59.83	-36.37
2806.00	57.57	-37.13
2812.00	54.72	-38.48
2818.00	52.43	-39.27
2900.00	50.06	-40.14
2906.00	47.91	-40.79
2912.00	44.88	-42.32
2918.00	42.58	-43.12
3000.00	40.14	-44.06
3006.00	37.26	-45.44

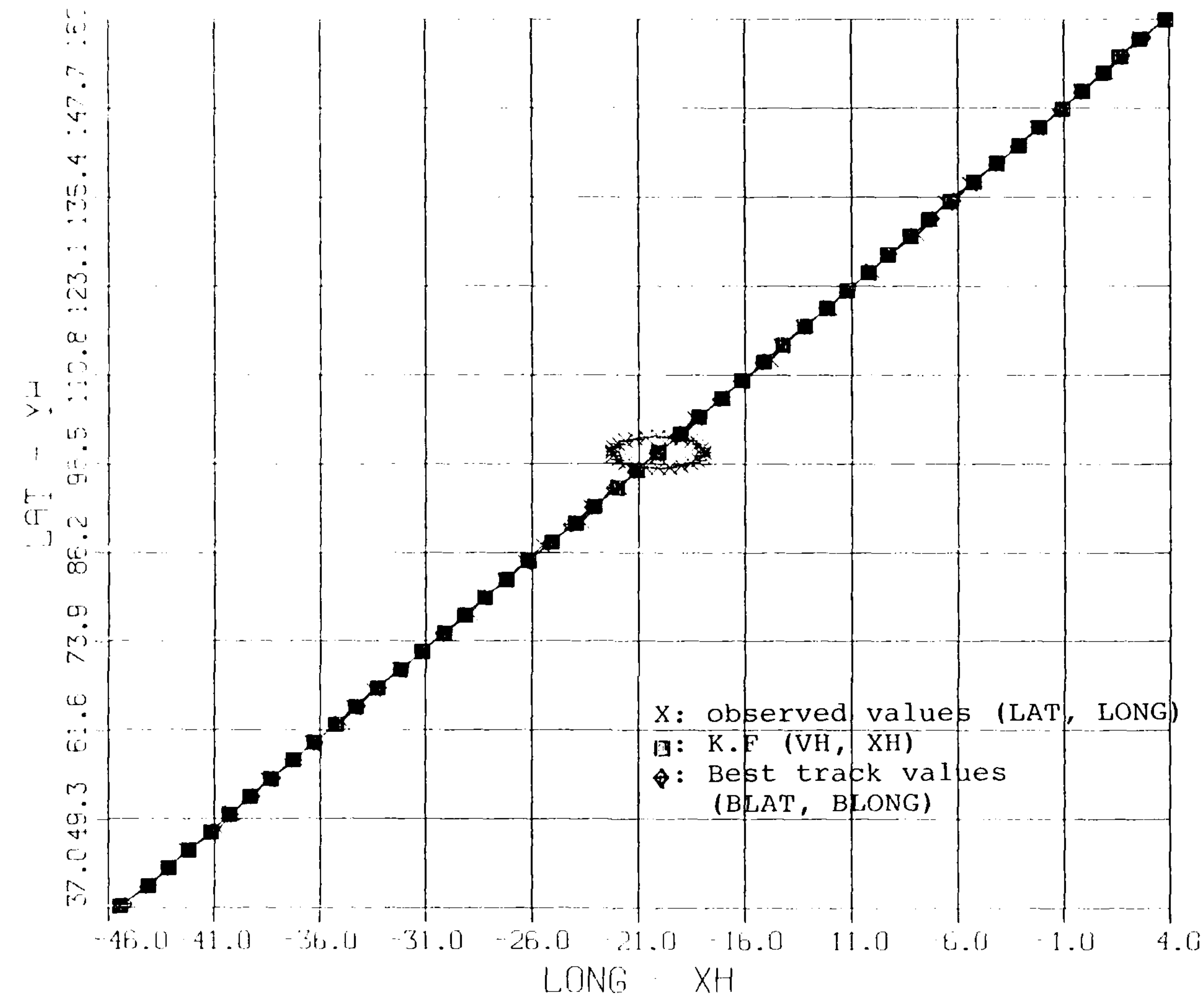


Figure 11 Trajectory of Fictitious Storm

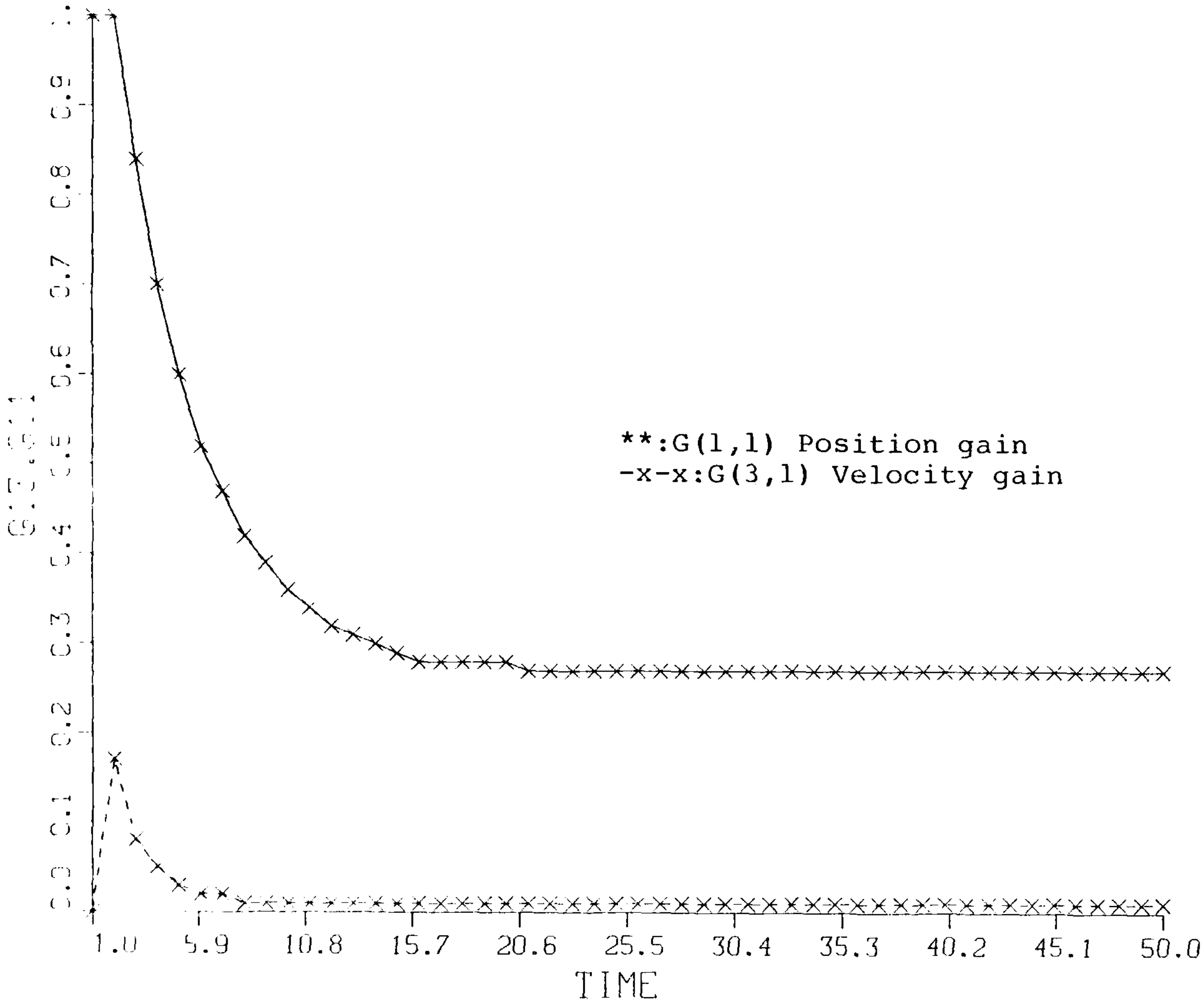


Figure 12 Gains of Position and Velocity

TABLE 7

ERROR VALUES IN TERMS OF LATITUDE AND LONGITUDE

JTIME	EB1	EB2	EUB1	EUB2
720624	-0.09	-0.06	0.0	0.0
720630	-0.20	-0.20	-0.20	-0.20
720636	-0.12	-0.13	-0.08	-0.08
720642	0.11	0.10	0.21	0.21
720648	0.10	0.09	0.03	0.03
720654	0.15	0.13	0.13	0.13
720660	0.09	0.07	-0.05	-0.05
720666	0.11	0.10	0.11	0.11
720672	0.08	0.07	-0.01	-0.01
720678	-0.02	-0.03	-0.24	-0.24
720684	-0.14	-0.15	-0.38	-0.38
720690	-0.15	-0.16	-0.11	-0.11
720696	-0.02	-0.03	0.31	0.31
720702	-0.05	-0.06	-0.13	-0.13
720708	0.02	0.01	0.17	0.17
720714	0.00	-0.00	-0.05	-0.05
720720	0.06	0.05	0.19	0.19
720726	0.02	0.02	-0.10	-0.10
720732	-0.02	-0.02	-0.13	-0.13
720738	0.10	0.10	0.42	0.42
720744	0.10	0.09	0.03	0.03
720750	0.12	0.12	0.15	0.15
720756	0.05	0.05	-0.17	-0.17
720762	0.17	0.17	0.46	0.46
720768	0.11	0.11	-0.10	-0.10
720774	0.12	0.12	0.15	0.15
720780	0.18	0.18	0.29	0.29
720786	0.12	0.12	-0.06	-0.06
720792	0.23	0.23	0.51	0.51
720798	0.10	0.10	-0.32	-0.32
720804	0.02	0.02	-0.15	-0.15
720810	-0.02	-0.02	-0.10	-0.10
720816	-0.03	-0.03	0.0	0.0
720822	0.04	0.04	0.28	0.28
720828	0.05	0.05	0.06	0.06
720834	0.04	0.04	0.0	0.0
720840	0.01	0.01	-0.04	-0.04
720846	-0.07	-0.07	-0.28	-0.28
720852	-0.12	-0.12	-0.21	-0.21
720858	-0.04	-0.04	0.25	0.25
720864	-0.08	-0.08	-0.17	-0.17
720870	-0.05	-0.05	0.07	0.07
720876	-0.12	-0.12	-0.28	-0.28
720882	-0.11	-0.11	-0.07	-0.07
720888	-0.08	-0.08	0.06	0.06
720894	0.05	0.05	0.41	0.41
720900	0.02	0.02	-0.12	-0.12
720906	0.04	0.04	0.08	0.08
720912	0.03	0.03	0.14	0.14
720918	-0.18	-0.18	-0.24	-0.24

TABLE 7

ERROR VALUES IN TERMS OF LATITUDE AND LONGITUDE

IDate	E01	E02	E0L1	E0L2
720624	-0.08	-0.06	0.0	0.0
720630	-0.20	-0.23	-0.20	-0.20
720636	-0.12	-0.13	-0.08	-0.08
720642	0.11	0.10	0.21	0.21
720648	0.10	0.09	0.03	0.03
720654	0.15	0.13	0.13	0.13
720660	0.09	0.07	-0.05	-0.05
720666	0.11	0.10	0.11	0.11
720672	0.08	0.07	-0.01	-0.01
720678	-0.02	-0.03	-0.24	-0.24
720684	-0.14	-0.15	-0.38	-0.38
720690	-0.15	-0.16	-0.11	-0.11
720696	-0.02	-0.03	0.31	0.31
720702	-0.05	-0.06	-0.13	-0.13
720708	0.02	0.01	0.12	0.12
720714	0.07	-0.00	-0.05	-0.05
720720	0.06	0.05	0.19	0.19
720726	0.02	0.02	-0.10	-0.10
720732	-0.02	-0.02	-0.13	-0.13
720738	0.10	0.10	0.42	0.42
720744	0.10	0.09	0.03	0.03
720750	0.12	0.12	0.15	0.15
720756	0.05	0.05	-0.17	-0.17
720762	0.17	0.17	0.46	0.46
720768	0.11	0.11	-0.10	-0.10
720774	0.12	0.12	0.15	0.15
720780	0.14	0.14	0.20	0.20
720786	0.12	0.12	-0.05	-0.05
720792	0.23	0.23	0.51	0.51
720798	0.10	0.10	-0.32	-0.32
720804	0.02	0.02	-0.15	-0.15
720810	-0.02	-0.02	-0.10	-0.10
720816	-0.03	-0.03	0.0	0.0
720822	0.04	0.04	0.28	0.28
720828	0.05	0.05	0.08	0.08
720834	0.04	0.04	0.0	0.0
720840	0.01	0.01	-0.04	-0.04
720846	-0.07	-0.07	-0.23	-0.23
720852	-0.12	-0.12	-0.21	-0.21
720858	-0.04	-0.04	0.25	0.25
720864	-0.08	-0.08	-0.17	-0.17
720870	-0.05	-0.05	0.07	0.07
720876	-0.12	-0.12	-0.23	-0.23
720882	-0.11	-0.11	-0.07	-0.07
720888	-0.08	-0.08	0.06	0.06
720894	0.05	0.05	0.41	0.41
720900	0.02	0.02	-0.12	-0.12
720906	0.04	0.04	0.08	0.08
720912	0.03	0.03	0.14	0.14
720918	-0.18	-0.18	-0.24	-0.24

```
      DO 25 I=1,L
   25 WRITE(8,90)(W(I,J),J=1,L)
      WRITE(8,140)
      GOTO 940
   49 DO 1115 I=1,L
 1115 WRITE(9,91)(W(I,J),J=1,L)
      WRITE(8,190)
  190 FORMAT(5X,'ENTER THE ORDER OF H, I.E. M=?')
      READ(5,40)M
      WRITE(8,195)
  195 FORMAT(5X,'ENTER THE ELEMENTS OF THE OBSERVATION
     1 MATRIX--H.')
      DO 7 I=1,M
      DO 7 J=1,N
      WRITE(8,200)I,J
  200 FORMAT(5X,'H(',I1,',',I1,')=')
      READ(5,70)H(I,J)
    7 CONTINUE
      WRITE(8,210)
  210 FORMAT('0',5X,'THE H MATRIX (OBSERVATION MATRIX)')
      DO 8 I=1,M
    8 WRITE(8,90)(H(I,J),J=1,N)
 1200 WRITE(8,100)
  920 READ(5,110)IAN
      IF(IAN.EQ.12)GOTO 39
      IF(IAN.NE.17)GOTO 1200
      WRITE(8,120)
      READ(5,130)I,J
      WRITE(8,200)I,J
      READ(5,70)H(I,J)
      WRITE(8,210)
      DO 9 I=1,M
    9 WRITE(8,90)(H(I,J),J=1,N)
      WRITE(8,140)
```

```
      WRITE(8,180)
      DO 6 I=1,N
    6 WRITE(8,90)(DEL(I,J),J=1,I)
      WRITE(8,140)
      GOTO 910
   29 WRITE(9,180)
      DO 1112 I=1,N
 1112 WRITE(9,91)(DEL(I,J),J=1,I)
      WRITE(8,410)
  910 WRITE(8,300)
  300 FORMAT(5X,'ENTER THE ELEMENTS OF THE COV OF W')
C
C NOTE W  IS CONSTRAINED TO (1,1) B KK 1710 & 1450
      W(1,1)=.000001
C
   76 DO 21 I=1,L
      DO 21 J=1,L
      WRITE(8,310)I,J
  310 FORMAT(5X,'W(',I1,',',I1,')=')
      READ(5,70)W(I,J)
   21 CONTINUE
   77 WRITE(8,321)
      WRITE(9,321)
  321 FORMAT('0',5X,'THE COV OF W')
   78 DO 22 I=1,L
   22 WRITE(8,90)(W(I,J),J=1, L)
 1400 WRITE(8,100)
  140 READ(5,110)IAN
      IF(IAN.EQ.IZ)GOTO 89
      IF(IAN.NE.IY)GOTO 1400
      WRITE(8,120)
      READ(5,130)I,J
      WRITE(8,310)
      READ(5,70)W(I,J)
      WRITE(8,321)
```

```
     +1X,'KALMAN FILTER PROGRAM')
      WRITE(8,142)(NAME(I),I=1,5)
  142 FORMAT(6X,'PROBLEM IDENTIFICATION:',5X,5A4)
      WRITE(8,143)
  143 FORMAT('0',70('*'))
      WRITE(8,80)
      DO 1111 I=1,N
 1111 WRITE(8,91)(PHI(I,J),J=1,N)
      WRITE(8,410)
      WRITE(8,150)
  150 FORMAT(5X,'ENTER THE DIMENSION OF  RANDOM INPUT
     1 VECTOR  (N)')
      READ(5,40)L
      WRITE(8,160)
  160 FORMAT(5X,'ENTER THE ELEMENTS OF THE DISTRIBUTION',
     +1X,'MATRIX--GAMMA.')
      DO 4 I=1,N
      DO 4 J=1,L
      WRITE(8,170)I,J
  170 FORMAT(5X,'GAMMA(',I1,',',I1,')=')
      READ(5,70)DEL(I,J)
    4 CONTINUE
      WRITE(8,180)
  180 FORMAT('0',5X,'THE GAMMA MATRIX  (DISTRIBUTION
     1 MATRIX)')
      DO 5 I=1,N
    5 WRITE(8,90)(DEL(I,J),J=1,L)
 1100 WRITE(8,100)
  910 READ(5,110)IAN
      IF(IAN.EQ.IZ)GOTO 29
      IF(IAN.NE.IY)GOTO 1100
      WRITE(8,120)
      READ(5,130)I,J
      WRITE(8,170)I,J
      READ(5,70)DEL(I,J)
```

```
    1 CONTINUE
      WRITE(8,30)
   30 FORMAT('0',5X,'THE PHI MATRIX (TRANSITION MATRIX)')
      DO 2 I=1,N
    2 WRITE(8,90) (PHI(I,J),J=1,N)
   90 FORMAT(1P7E11.4)
   91 FORMAT(1P7E11.4)
C******************************
C
C     GO TO 883
C******************************
 1000 WRITE(8,100)
  100 FORMAT(5X,'DO YOU WANT TO CHANGE ANY ELEMENT OF THE
     1 MATRIX?')
  900 READ(5,110)IAN
  110 FORMAT(A3)
      IF(IAN.EQ.IZ)GOTO 19
      IF(IAN.NE.IY)GOTO 1000
      WRITE(8,120)
  120 FORMAT(5X,'WHICH ELEMENT OF THE MATRIX DO YOU WANT
1 TO CHANGE?',/,
     +5X,'ENTER AS IJ; WHERE I IS THE ROW AND J IS THE
     1 COLUMN.')
      READ(5,130)I,J
  130 FORMAT(2I1)
      WRITE(8,60)I,J
      READ(5,70)PHI(I,J)
      WRITE(8,30)
      DO 3 I=1,N
    3 WRITE(8,90) (PHI(I,J),J=1,N)
      WRITE(8,140)
  140 FORMAT(5X,'ANY OTHER CHANGES?')
      GOTO 900
   19 WRITE(9,141)
  141 FORMAT('1',5X,'DISCRETE TIME',
```

```
      WRITE(9,90) (A(I,J),J=1,N)
 7780 WRITE(8,90) (A(I,J),J=1,N)
C     WRITE(8,4422)
C4422 FORMAT(5X,'  ENTER THE DIMENSION  OF  W    ')
C
C     W IS CONSTRAINED TO (1,1) HERE & D KK 045508 02700
C
C
C     READ(5,4423)I
C4423 FORMAT(I1)
      I=1
      WRITE(8,7781)
 7781 FORMAT(5X,' ENTER  THE  B MATRIX  ')
      DO 7782 I=1,N
      DO 7782 J=1,L
      WRITE(8,7783)I,J
 7783 FORMAT(5X,'B(',I1,',',I1,')=')
      READ(5,7778)B(I,J)
 7782 CONTINUE
      WRITE(9,7784)
      WRITE(8,7784)
 7784 FORMAT(5X,'  THE B MATRIX   ')
      DO 7785 I=1,N
      WRITE(9,90) (B(I,J),J=1,L)
 7785 WRITE(8,90) (B(I,J),J=1,L)
      GO TO 9010
   15 WRITE(8,50)
   50 FORMAT(5X,'ENTER THE ELEMENTS OF THE TRANSITION
     1 MATRIX--PHI')
      DO 1 I=1,N
      DO 1 J=1,N
      WRITE(8,60)I,J
   60 FORMAT(5X,'PHI(',I1,',',I1,')=')
      READ(5,70)PHI(I,J)
   70 FORMAT(F10.0)
```

```
      ND=12
      MD=12
      LD=12
      DO 7898 I=1,N
      Z(I)=0.
      DO 7898 J=1,N
      HI(I,J)=0.
      HI(I,I)=1.
      PHI(I,J)=0.
      PHI(I,I)=1.
      H(I,J)=0.
      B(I,J)=0
 7898 A(I,J)=0.
      WRITE(8,410)
      WRITE(8,7771)
 7771 FORMAT(5X,'DO YOU WANT TO COMPUTE PHI S GAMMA ON
     1    LINE FROM',
     +1X,'A S B?')
      READ(5,7772)IAN
 7772 FORMAT(A3)
      IF(IAN .EQ.IPH)GO TO 7773
      IF(IAN .NE.IPH)GO TO 15
 7773 WRITE(8,7774)
 7774 FORMAT(//,5X,'ENTER THE A MATRIX')
      DO 7775 I=1,N
      DO 7775 J=1,N
      WRITE(8,7776)I,J
 7776 FORMAT(5X,'A(',I1,',',I1,')=')
      READ(5,7778)A(I,J)
 7778 FORMAT(F10.3)
 7775 CONTINUE
      WRITE(9,7779)
      WRITE(8,7779)
 7779 FORMAT(5X,'  THE A MATRIX    ')
      DO 7780 I=1,N
```

```
      WRITE(8,27)
   27 FORMAT(/2X,'                                                  '/,
     *'*************************************************'/,
     *2X,'DO NOT USE PA1 KEY TO EXIT PROGRAM UNLESS
     1 YOU WANT'/,
     *2X'TO LOSE YOUR INPUT FILE THAT IS PRINTED ON
     1 UNIT 4.'/,
     *'*************************************************'/,
     *2X'FOR CHECK OUT ACTIVATE THE GO TO 888 STATEMENT
     1 ON'/,
     *2X'LINE 135 TO END WITHOUT ERROR'/,
     *'*************************************************'/,
     *2X,'DO YOU WANT TO DEVELOP THE INPUT FILE (Y/N)?')
C
      READ (5,33)IANS
   33 FORMAT (A4)
C
      IF (IANS .NE. IYEN1) GO TO 9999
C
C
C
C
 7777 FORMAT(1H1)
      WRITE(8,1234)
 1234 FORMAT('ALL INPUT SHOULD BE FLOATING POINT EXCEPT ')
      WRITE(8,1235)
 1235 FORMAT(' WHERE OTHERWISE SPECIFIED ',/)
 1016 FORMAT(1H0,5X,5H K = ,I3,/5X,5HGAINS)
 1902 WRITE(8,10)
   10 FORMAT(/,5X,' THE DISCRETE KALMAN FILTER')
      WRITE(8,30)
   30 FORMAT(5X,'      ENTER THE ORDER OF THE SYSTEM
     1 (UP TO 8).')
      READ(5,40) N
   40 FORMAT(I1)
```

```
C
C
C     PKK(I,J) = P(K/K), (COV ERROR AT K GIVEN K SAMPLES)
C
C
C PKKM1(I,J) = P(K/K-1),(COV ERROR AT K GIVEN K-1
CSAMPLES)
C   N = NUMBER OF ROWS, M = NUMBER OF COL.,OBS.
C,ND AND MD AR
C           NN = NUMBER OF ITERATIONS OF THE FILTER
C
C
C ********* NEW NEWS ************************************
       WRITE(6,7171)
 7171 FORMAT (/,2X,'*************NEW NEWS**************'/,
     *2X,'THE STORM TRACK INPUT VALUES ARE AVIALABLE'/,
     *2X,'THESE ARE ECHO PRINTED TO THE TERMINAL AT THE'/,
     *2X,'BEGINNING OF THE PROGRAM -- ALSO SEE LISTING'/,
     *2X,'IF THE PROGRAM ENDS NORMALLY AN INPUT FILE
     1 WILL'/,
     *2X,'BE PRODUCED THAT MAY BE PRINTED OR USED
     1  FOR INPUT'/,
     *2X,'USING THE SAME FORMAT STATEMENTS TO READ AS
     1   WERE'/,
     *2X,'USED TO WRITE ON UNIT 4.  SO WITH A FEW
     1  MODIFICATIONS'/,
     *2X,'THE BRANCH AROUND THE INPUT CAN BE USED.
     1 PRESENTLY'/,
     *2X,'A FILE OF THE INPUT DATA IS BEING PRODUCED AND
     1 WILL'/,
     *2X,'BE FOUND AS --K OUTPUT A-- ON YOUR A - DISK.')
C
C  **** THE FIRST QUESTION TO THE TERMINAL --ASKS IF
C        AN INPUT FILE IS TO BE USED
C
```

```
  373 FORMAT (2X,7F10.2)
C      WRITE(9,373) (TIME(I),LAT(I),LONG(I),BTIME(I)
      1 ,BLAT(I),
C     *      BLONG(I),BWIND(I), I=1,NZ)
C      WRITE(9,374) (TIME(I),LAT(I),LONG(I),BTIME(I)
C     1 ,BLAT(I),
C     *      PCN(I), I=1,NZ)
C 374 FORMAT (2X,6F10.2)
C


C      THIS PROGRAM COMPUTES THE FOLLOWING KALMAN FILTER
C
C                                                 -1
C         G(K) = P(K/K-1)*HT*(H*P(K/K-1)*HT+R)
C
C
C         P(K/K) = (I-G(K)*H)*P(K/K-1)
C
C
C         P(K/K-1) = PHI*P(K/K-1)*PHIT+Q
C
C
C Q(I,J) DEFINES THE COVARIANCE OF THE PER SAMPLE RANDOM
C             EXCITATION OF THE PROCESS
C
C
C R(I,J) DEFINES THE RANDOM (GAUSSIAN) MEASUREMENT NOISE
C             WHICH IS ADDED TO THE OBSERVABLE SIGNALS
C
C
C         HI(I,J) IS THE IDENITY MATRIX
C
C
C II=K THE DISCRETE POINT IN TIME,THE STAGE OF THE PROCESS
```

```
      INTEGER IYEN1    /'Y'/
C
C NZ= NO. OF OBSERVED VALUES(SATELLITE)
C MZ=NO. OF BEST TRACK VALUES
C
      NZ=50
      MZ=50
      DO 132 I=1,4
      PKK(I,I) = 1000.
 132  PKKM1(I,I)=1000.
      W(1,1) = .000001
      W(2,2) = .000001
C READ OBSERVED VALUES
C
C     READ(2,11)(TIME(I),LAT(I),LONG(I),PCN(I),I=1,NZ)
   11 FORMAT (7X,F6.2,F3.1,1X,F4.1,1X,F1.0)
      READ(2,11)(TIME(I),LAT(I),LONG(I),PCN(I),I=1,NZ)

C
C READ BEST TRACK VALUES
C
      READ(3,14)(BTIME(J),BLAT(J),BLONG(J),BWIND(J)
     1 ,J=1,MZ)
   14 FORMAT (6X,F4.0,F4.1,1X,F4.1,2X,F3.0)
C     READ(3,11)(BTIME(J),BLAT(J),BLONG(J),PCN(J),J=1,MZ)
C  11 FORMAT (2X,4F10.2)
C  11 FORMAT (6X,F4.0,F4.1,1X,F4.1,2X,F3.0)

C ECHO VALUES  ****REMOVE NEXT THREE LINES TO ELIMINATE
C     ECHO PRINT CHECK ***********
C
      WRITE(8,373) (TIME(I),LAT(I),LONG(I),BTIME(I)
     1 ,BLAT(I),
     *      BLONG(I),PCN(I), I=1,NZ)
C     WRITE(8,373) (TIME(I),LAT(I),LONG(I),PCN(I),I=1,NZ)
```

APPENDIX A

COMPUTER ALGORITHM

```
C      KALMAN FILTER
       DIMENSION HI(12,12),H(12,12),F(12,12),G(12,12)
      1 ,PHIT(12,12),
      *   Q(12,12), G31(120),PKK(12,12),PKKM1(12,12)
      1 ,G11(120),G22(120)
       DIMENSION IREAD(10),IWRITE(10),Y(120),YH(120)
      1 ,XH(120),G44(120)
       DIMENSION DEL(12,12),A(12,12),B(12,12),D1(12,12)
      1   ,D2(12,12)
       DIMENSION Z(12),E(12),GE(12),XHP(120),YHP(120)
      1 ,EY(120),EX(120)
       DIMENSION DELI(12,12),PHI(12,12),TI(120),XKKM6(12)
C
C
       DIMENSION NAME(5),D(12,12),XP(25),YP(25),KTOLHS(120),
      *W(12,12),F(12,12),AI(12,12),ZKKM1(12),IK(120)
      1 ,PKK22(120),P22(120),
      *FT(12,12),ET(12,12),XKK(12),XKKM1(12),ZXY(120)
      1 ,EOB1(120)
       DIMENSION TIME(120),LAT(120),LONG(120),PTIME(120)
      1 ,BLAT(120),
      *BLONG(120),BWIND(120),XK46(120),ZK46(120),PCN(120)
      1 ,EOB2(120)
       INTEGER IKM72(100),IKM48(100),IKM24(100),
      * IKM12(100),IKM6(100),IKP6(100),IKP12(100)
      1 ,IKP24(100),
      * IKP48(100),IKP72(100)
       DATA IQIT/1H$/,IY/'Y'/,IZ/'N'/,IPH/'Y'/
       REAL LAT,LONG
       INTEGER IYFNC    /'Y'/
```

VI. CONCLUSIONS

The K.F approach to estimate the storm's location appears to be very accurate. That comes from the comparison with the meteorologist's analysis results. Concerning also the fact that the latter was performed after the storm's occurence one can see the advantages of the K.F algorithm. During operation of the filter the actual residual sequence $[\underline{z}(k)-\hat{\underline{z}}(k|k-1)]$ is compared to the gate $[\sqrt{\underline{P}(k|k-1)+\underline{R}(k)}]$ which actually is the square root of the residual covariance. Being a white Gaussian sequence, the residual is bounded by this gate.

When the gate is exceeded the model is determined invalid within the filter and a modification in terms of Q and G takes place to adapt to the situation. At this point, and if the excess occurs in only one component of the vector residual process, one can further deduce that the measuring device generating the particular component is the source of difficulty (a sensor failure).

The error ellipsoids of the process also give insight into the filter performance in a more general case, referring to many sensors with a greater variety of uncertainties, the adaptive K.F algorithm could be a very advantageous approach.

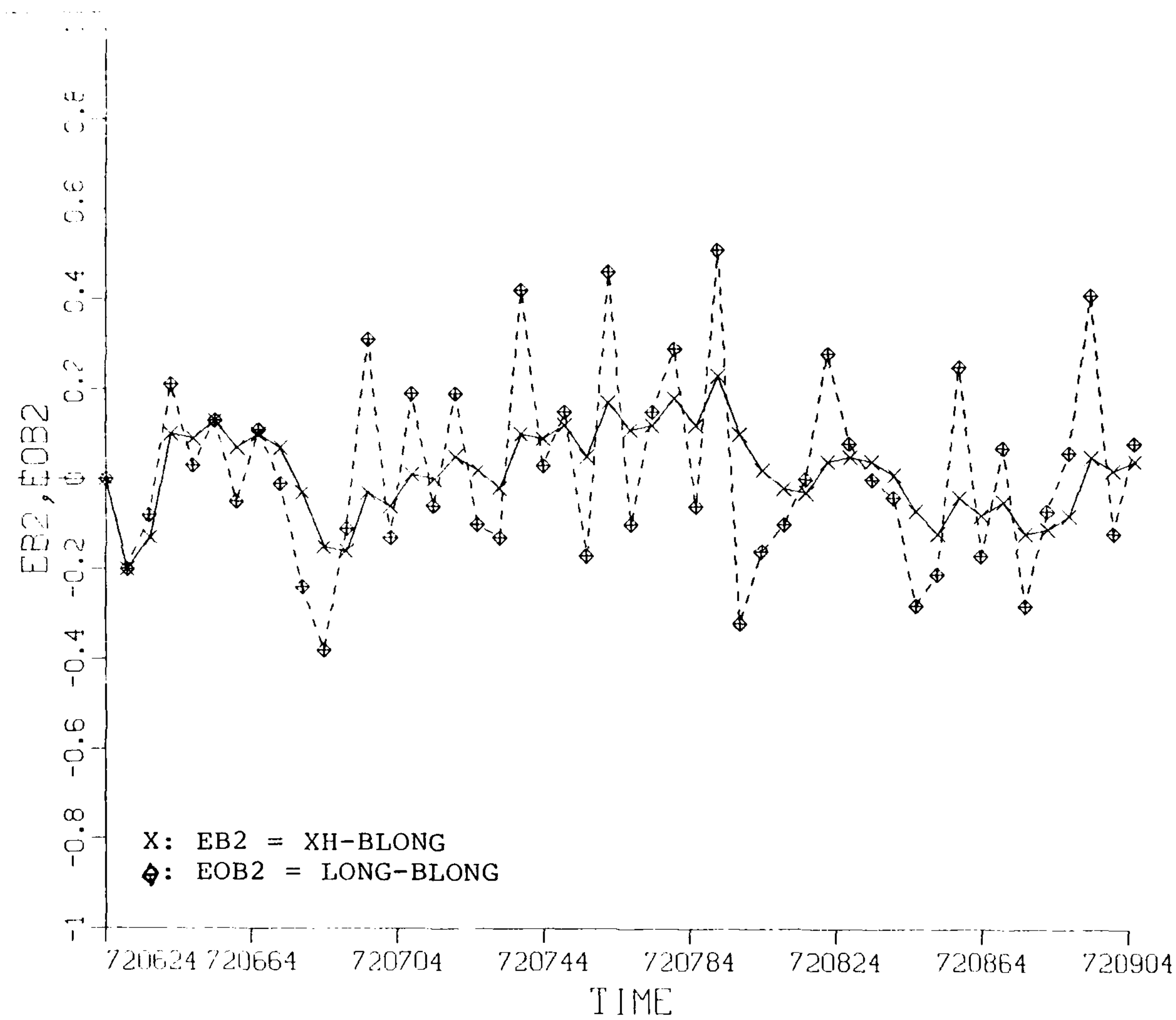


Figure 14 Longitude Errors

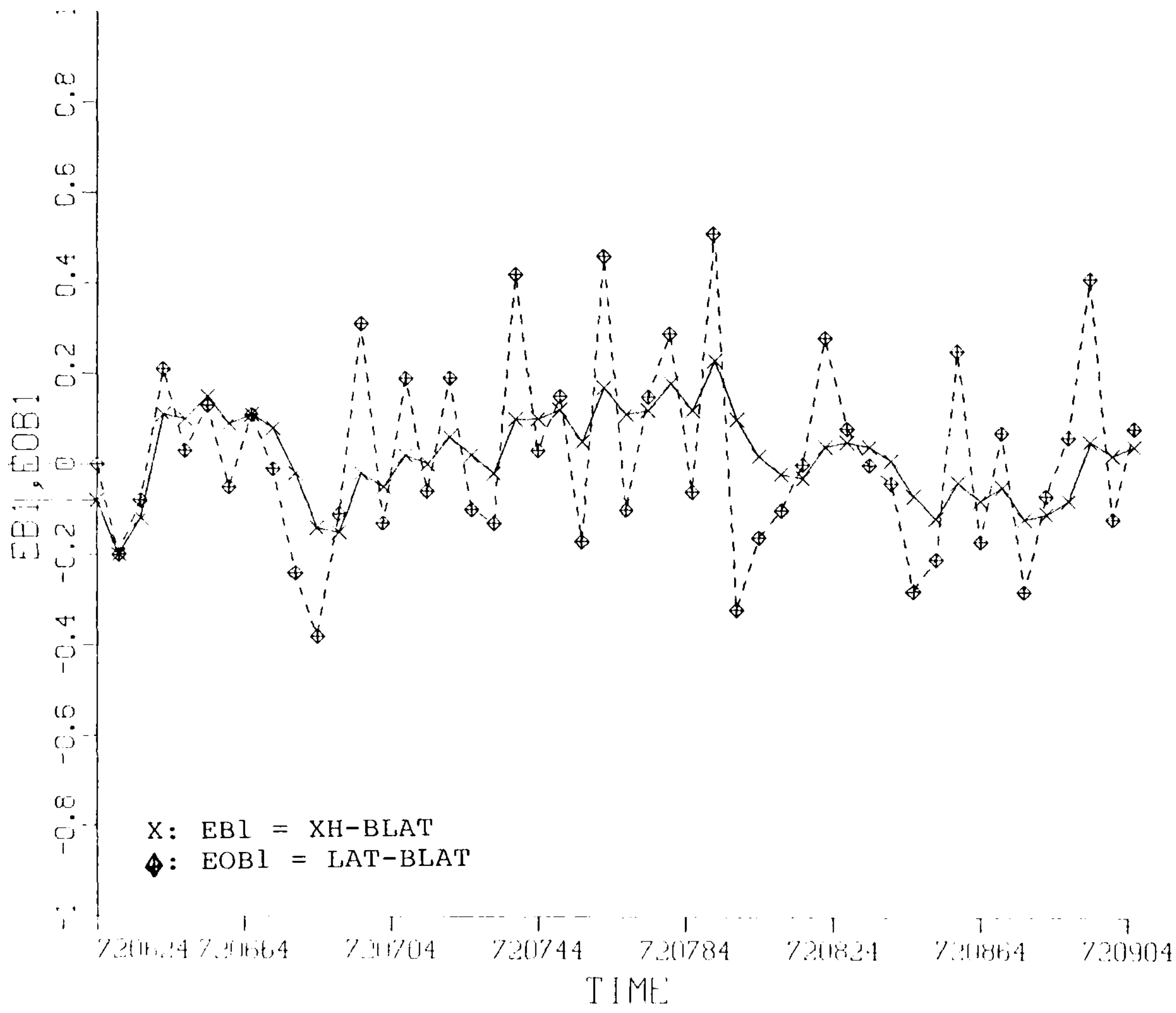


Figure 13 Latitude Errors

```
      GOTO 920
   39 WRITE(9,210)
      DO 1113 I=1,M
 1113 WRITE(9,91)(R(I,J),J=1,M)
      WRITE(9,140)
      WRITE(8,270)
  270 FORMAT(5X,'ENTER THE ELEMENTS OF THE MEASUREMENT
     1 NOISE',
     +1X,'COVARIANCE MATRIX--R  ')
   72 DO 116 I=1,M
      DO 116 J=1,M
      WRITE(8,280)I,J
  280 FORMAT(5X,'R(',I1,',',I1,')=')
      READ(5,70)R(I,J)
  116 CONTINUE
      WRITE(8,290)
      WRITE(9,290)
  290 FORMAT('0',5X,'THE R MATRIX (MEASUREMENT',
     +1X,'NOISE COVARIANCE MATRIX)')
   74 DO 17 I=1,M
   17 WRITE(8,90)(R(I,J),J=1,M)
 1300 WRITE(8,100)
  930 READ(5,110)IAN
      IF(IAN.EQ.IZ)GOTO 79
      IF(IAN.NE.IY)GOTO 1300
      WRITE(8,120)
      READ(5,130)I,J
      WRITE(8,280)I,J
      READ(5,70)R(I,J)
      WRITE(8,290)
      DO 18 I=1,M
   18 WRITE(8,90)(R(I,J),J=1,M)
      WRITE(8,140)
      GOTO 930
   79 DO 1114 I=1,M
```

```
1114 WRITE(9,91)(P(I,J),J=1,N)
     WRITE(8,410)
     WRITE(8,330)
 330 FORMAT(//,5X,'ENTER      PKKM1(1/0)                ')
  42 DO 34 I=1,N
     DO 34 J=1,N
     WRITE(8,340)I,J
 340 FORMAT(5X,'PKKM1(',I1,',',I1,')=')
     READ(5,70)PKKM1(I,J)
  34 CONTINUE
  43 WRITE(8,351)
     WRITE(9,351)
 351 FORMAT('0',5X,'PKKM1(1/0)        ')
  44 DO 35 I=1,N
  35 WRITE(8,90)(PKKM1(I,J),J=1,N)
1500 WRITE(8,100)
 950 READ(5,110)IAN
     IF(IAN.EQ.IZ)GOTO 51
     IF(IAN.NE.IY)GOTO 1500
     WRITE(8,120)
     READ(5,130)I,J
     WRITE(8,340)I,J
     READ(5,70)PKKM1(I,J)
     WRITE(8,351)
     DO 36 I=1,N
  36 WRITE(8,90)(PKKM1(I,J),J=1,N)
     WRITE(8,140)
     GOTO 950
  51 DO 1116 I=1,N
1116 WRITE(9,91)(PKKM1(I,J),J=1,N)
     WRITE(9,143)
     WRITE(8,246)
 246 FORMAT(5X,'ENTER THE NUMBER OF THE POINTS TO BE
    1 PERFORMED.',
    +/,5X,'(<100) THIS IS AN INTEGER VALUE')
```

```
      READ(5,247)NN
 247  FORMAT(I2)
      IT=(TIME(1)-1400)*(.01)
      IS=(TIME(2)-1400)*(.01)
      I=24*IT
      S=24*IS
      T=TIME(1)-IT*100-1400+I
      S=TIME(2)-IS*100-1400+S
      DI=S-T
3753  CONTINUE
      JK=1
      WRITE(8,7865)JK
7865  FORMAT(I4)
      WRITE(8,511)N,M,L,ND,MD,LD,NN,DI
 511  FORMAT(2X,2HN=,I5,5X,2HM=,I5,5X,2HL=,I5,5X,3HND=
     1 ,I5,5X,3HMD,I5,
     *5X,3HLD=,
     6I5,5X,3HNN=,I5,5X,3HDI=,F10.4)
      WRITE(8,533)
 533  FORMAT(/'   MATRIX R ')
C
      DO 3017 I=1,M
3017  WRITE(8,90) (R(I,J),J=1,M)
      WRITE(8,544)
 544  FORMAT(/'   MATRIX Q ')
      DO 3018 I=1,N
3018  WRITE(8,90) (Q(I,J),J=1,N)
      WRITE(8,555)
 555  FORMAT(/'   MATRIX PKKM1')
      DO 3019 I=1,N
3019  WRITE(8,90) (PKKM1(I,J),J=1,N)
C     IF(IANS.NE.IPH) GO TO 6789
```

```
      IT=(TIME(1)-1400)*(.01)
      IS=(TIME(2)-1400)*(.01)
      T=24*IT
      S=24*IS
      T=TIME(1)-IT*100-1400+T
      S=TIME(2)-IS*100-1400+S
      DT=S-T
      JK=2
      WRITE(8,7865)JK
 9753 CONTINUE
      WRITE(8,7878)DT
 7878 FORMAT(5X,F10.3)
      CALL PHIDEL(DT,N,L,A,B,PHI,DEL,D1,D2,ND,MD,LD)
 6789   WRITE(8,666)
  666   FORMAT(/'      PHI        ')
        DO 3020 I=1,N
 3020   WRITE(8,90) (PHI(I,J),J=1,N)
        WRITE(8,777)
  777   FORMAT(/'      DEL        ')
        DO 3021I=1,N
 3021   WRITE(8,90) (DEL(I,1),J=1,L)
        CALL TRANS(DEL,N,1,DELT,ND,MD)
        CALL PROD(DEL,DELT,N,1,N,Q  ,ND,MD,LD)
        CALL CONST(W(1,1),Q,N,N,Q,ND,MD)
        WRITE(8,544)
        DO 3025 I=1,N
 3025   WRITE(8,90) (Q(I,J),J=1,N)
        WRITE(8,444)
  444   FORMAT(/'          H     ')
        DO 3026 I=1,M
 3026   WRITE(8,90) (H(I,J),J=1,N)
       JK=3
       WRITE(8,7865)JK
```

```
      WRITE(4,5111)N,M,L,ND,MD,LD,NN,DI
5111  FORMAT(7I4,F10.4)
      DO 5327 I=1,N
5327  WRITE(4,90) (A(I,J),J=1,N)
      DO 5328 I=1,N
5328  WRITE(4,90) B(I,1)
      DO 5329 I=1,N
5329  WRITE(4,90)(Q(I,J),J=1,N)

      DO 5330 I=1,N
5330  WRITE(4,90) (EKKM1(I,J),J=1,N)
      DO 6327 I=1,M
6327  WRITE(4,90) (B(I,J),J=1,N)
      DO 6927 I=1,N
6927  WRITE(4,90) (HI(I,J),J=1,N)
      DO 6328 I=1,M
6328  WRITE(4,90) (R(I,J),J=1,M)
        WRITE(8,7777)
9999    CONTINUE
      IF(IANS.EQ.IYEN1)GO TO 7234
      WRITE(6,1928)IANS
1928  FORMAT(5X,' XXXXXXXXXXXXXX ',A4)
      JK=33
      WRITE(8,7865)JK
      READ(4,5111)N,M,L,ND,MD,LD,NN,DI
      JK=99
      WRITE(8,7865)JK
      WRITE(8,5111)N,M,L,ND,MD,LD,NN,DI
      JK=4
      WRITE(8,7865)JK
      DO 7235 I=1,N
7235  READ(4,91) (A(I,J),J=1,N)
      JK=4321

      WRITE(8,7865)JK
      DO 7236 I=1,N
```

```
7236 READ(4,91) E(I,1)
     DO 7238 I=1,N
7238 READ(4,91) (Q(I,J),J=1,N)
     DO 7239 I=1,N
     READ(4,91) (PKKM1(I,J),J=1,N)
7239 WRITE(8,90) (PKKM1(I,J),J=1,N)
     DO 7326 I=1,M
7326 READ(4,91) (H(I,J),J=1,N)
     DO 7329 I=1,N
7329 READ(4,91) (HI(I,J),J=1,N)
     DO 7327 I=1,M
7327 READ(4,91) (R(I,J),J=1,M)
     II=(TIME(1)-1500)*(.01)
     IS=(TIME(2)-1500)*(.01)
     T=24*II

     S=24*IS
     T=TIME(1)-II*100-1500+T
     S=TIME(2)-IS*100-1500+S
     DT=S-T
5253 CONTINUE
     W(1,1)=.000001
     JK=5
     WRITE(6,7555) JK
     CALL TRANS([illegible])
     CALL PROD([illegible])
     CALL CONST([illegible])
     CALL PHIDFI([illegible],S1,S2,ND,MD,ID)
7234 CONTINUE
     DDT=DT
     WRITE(6,9162)
9162 FORMAT(5X,' [illegible] 2222 ' )
C [illegible]
     EB=1500+II*100
     KKK=1
```

```
      DO 2222  K = 1,NN
C
C     *** CALLS SUBROUTINE TO CALCULATE JULIAN TIME
C     *** FOR EVERY STORM POSITION AND EVERY 6 HOURS
      ITIME=INT(TIME(K))
      IDAY=ITIME/100
      IHOUR=ITIME-(IDAY*100)
      IF(IHOUR.EQ.0)IHOUR=24
      MOD6=MOD(IHOUR,6)
      CALL JUTIME (ITIME,JULHR)
C     IF(MOD6.EQ.0)GOTO 1986
      MTIME=INT(DD)
      MDAY=MTIME/100
      MDAY=MDAY*100
      MHOUR=MTIME-MDAY
      IF(MHOUR.GT.24)MTIME=ITIME
      CALL JUTIME (MTIME,MJULHR)
 1986 CONTINUE
      WRITE(8,1989)JULHR,ITIME
 1989 FORMAT(///,' JULIAN HOUR IS ',I9,' ,ACTUAL TIME IS:
     1  ',I6)
      WRITE(8,1984)MTIME,MJULHR
 1984 FORMAT(' MODULA 6 TIME=  ',I5,' ,CORRESPONDING
     1 JULIAN HR=',I9)
 1985 CONTINUE
C     *** END JULIAN TIME ROUTINE
C
C     *** CALCULATE MODULA 6  JULIAN TIME FOR IKP,IKM,KKP C
,KKM:
C
C     *** MODULA 6 FOR JULIAN TIME
C     IF(MHOUR.EQ.18)GOTO 3187
C
      IKM72(K)=MJULHR-72
      IKM48(K)=MJULHR-48
```

```
      IKM24(K)=MJULHR-24
      IKM12(K)=MJULHR-12
      IKM6(K)=MJULHR-6
      IKP6(K)=MJULHR+6
      IKP12(K)=MJULHR+12
      IKP24(K)=MJULHR+24
      IKP48(K)=MJULHR+48
      IKP72(K)=MJULHR+72
 3187 CONTINUE
C     *** END JULIAN TIME ROUTINE
C
      IF(PCN(K)-5.NE.0) GO TO 3133
      R(1,1)=.25
      R(2,2)=.25
      GO TO 3134
 3133 IF(PCN(K)-3.NE.0) GO TO 3135
      R(1,1)=.0625
      R(2,2)=.0625
      GO TO 3134
 3135 IF(PCN(K)-2.NE.0) GO TO 3134
      R(1,1)=.0312
      R(2,2)=.0312
 3134 CONTINUE
        WRITE(8,5445) TIME(K)
 5445   FORMAT(//////50X,5HTIME=,F10.4)
        WRITE(4,5445) TIME(K)
      WRITE(8,9313)K,BB,R(1,1),Q(1,1)
 9313 FORMAT(3X,'K=',I3,5X,'BB=',F8.2,5X,'R=',F7.4,5X,
     1  'Q(1,1)='F10.4)
      WRITE(4,9313)K,BB,R(1,1),Q(1,1)
      WRITE(8,313)    PCN(K),DT,W(1,1)
  313 FORMAT(3X,' PCN(K)=',F10.4,3X,'DT=',F6.2,10X,'W(1,1)
     1 =',F10.4)
      WRITE(4,313)    PCN(K),DT,W(1,1)
```

```
      DO 3129 I=1,N
      DO 3129 J=1,N
3129  Q(I,J) = 0.
      Q(1,1) = (DT**4/4)*W(1,1)
      Q(2,2) = (DT**4/4)*W(2,2)
      Q(3,3) = DT**2*W(2,2)
      Q(4,4) = DT**2*W(2,2)
      WRITE(8,799)
 799  FORMAT (/'    MATRIX  Q     ')
      DO 3123 I=1,N
3123  WRITE(8,90) (Q(I,J),J=1,N)
      W(1,1)=.000001
      W(2,2)=.000001
      CALL PHIDEL(DT,N,L,A,B,PHI,DEL,D1,D2,ND,MD,LD)
        WRITE(8,979)
 979    FORMAT(/'   PHI    ')
        WRITE(4,979)
      DO 3579 I=1,N
      WRITE(4,90) (PHI(I,J),J=1,N)
3579  WRITE(8,90) (PHI(I,J),J=1,N)
        CALL GAIN(PKK,PKKM1,Q,R,PHI,H,N,M,G,HT,ND,MD
     1      ,LD,K)
        WRITE(4,656)
        WRITE(8,656)
 656    FORMAT(/'   PKK    ')
        DO 3023 I=1,N
        WRITE(8,90) (PKK(I,J),J=1,N)
3023    WRITE(4,90) (PKK(I,J),J=1,N)
        CALL PROD(PHI,XKK,N,N,1,XKKM1,ND,MD,LD)
        CALL PROD(H,XKKM1,N,M,1,ZKKM1,ND,MD,M )
        WRITE(8,3810)
3810    FORMAT(/'     ZKKM1       ')
        WRITE(8,90) (ZKKM1(J),J=1,M)
        WRITE(4,90) (ZKKM1(J),J=1,M)
       WRITE(8,3819)
```

```
 3319    FORMAT(/' LAT(K),LONG(K)  ')
C        WRITE(8,90)LAT(K),LONG(K)
         Z(1)=LAT(K)
         Z(2)=LONG(K)
         WRITE(4,8811)
         WRITE(4,90) (Z(J),J=1,M)
         WRITE(8,8811)
         WRITE(8,90) (Z(J),J=1,M)
         CALL SUB(Z,ZKKM1,M,1,E,ND,MD)
         WRITE(8,5445) TIME(K)
C        WRITE(8,8810)
         WRITE(8,3029)
 3029    FORMAT(/ '    E     ************************** ')
         WRITE(8,90)( E(J),J=1,N)
        IF(K.LE.1) GO TO 2204
         GATE=(PKKM1(2,2)+R(1,1))**.5
         IF(ABS(E(2))-GATE .LT.0.) GO TO 2203
       G(2,2)=0.5*(1.2+G(2,2))
       W(2,2)=10000.*W(2,2)
       G(4,2)=0.5*(0.333+G(4,2))
C      PKKM1(1,1)=2*PKKM1(1,1)
C      PKKM1(2,2)=2*PKKM1(2,2)
C      PKKM1(3,3)=2*PKKM1(3,3)
C      PKKM1(4,4)=2*PKKM1(4,4)
C      WRITE(4,9192)GATE,E(2)
       WRITE(8,9192)GATE,E(2)
 2203 CONTINUE
       GATE=(PKKM1(1,1)+R(1,1))**.5
         IF(ABS(E(1))-GATE .LT.0.) GO TO 2204
       G(1,1)=0.5*(1.2+G(1,1))
       W(1,1)=10000.*W(1,1)
       G(3,1)=0.5*(0.333+G(3,1))
C      PKKM1(1,1)=2*PKKM1(1,1)
C      PKKM1(2,2)=2*PKKM1(2,2)
```

```
C       PKKM1(3,3)=2*PKKM1(3,3)
C       PKKM1(4,4)=2*PKKM1(4,4)
C       WRITE(4,9191)GATE,E(1)
        WRITE(8,9191)GATE,E(1)
 9191 FORMAT(9X,'ERROR GT GATE. GATE= ',F10.4,9X,'E(1)= '
     1   ,F10.4,'XXX')
 9192 FORMAT(9X,'ERROR GT GATE. GATE= ',F10.4,9X,'E(2)= '
     1  ,F10.4,'XXX')
C
C
C
C
C          G11(K)=G(1,1)
C          G31(K)=G(3,1)
C       DO 3022 I=1,N
C       WRITE(8,90)(PKKM1(I,J),J=1,N)
 3022 WRITE(4,90)(PKKM1(I,J),J=1,N)
 2204     WRITE(4,99)
          WRITE(8,99)
   99     FORMAT(/'    MATRIX G   ')
          DO 3024 I=1,N
          WRITE(8,90)(G(I,J),J=1,M)
 3024     WRITE(4,90)(G(I,J),J=1,M)
          WRITE(8,90)(ZKKM1(J),J=1,M)
          WRITE(8,8811)
          WRITE(4,8811)
 8811     FORMAT(/'       Z       ')
          WRITE(8,90)(Z(J),J=1,M)
          WRITE(4,90)(Z(J),J=1,M)
          CALL PROD(G,Z,N,M,1,GE,ND,MD,LD)
C         WRITE(4,90)(GE(J),J=1,N)
          CALL ADD(XKKM1,GE,N,1,XKK,ND,MD)
C         CALL PROD(PHI,XKK,N,N,1,XKKM1,ND,MD,LD)
          WRITE(8,8911)
          WRITE(4,8911)
```

```
 8811    FORMAT(/'     XKK              ')
         WRITE(8,90) (XKK(J),J=1,N)
         WRITE(4,90) (XKK(J),J=1,N)
         G11(K)=G(1,1)
         G31(K)=G(3,1)
C        P11(K)=PKKM1(1,1)
C        PK11(K)=PKK(1,1)
         YH(K)=XKK(1)
         XH(K)=XKK(2)
         KTIME=INT(TIME(K))
         RKTIME=FLOAT(KTIME)
         IF(RKTIME.NE.TIME(K))GOTO 8813
         KJULHR(K)=JULHR
C        EOB1(K)=YH(K)-LAT(K)
C        EOB2(K)=XH(K)-LONG(K)
 8813    CONTINUE
C        WRITE(4,8812)
C        WRITE(8,8812)
 8812 FORMAT(/'         XKKM1     ')
      WRITE(8,90) (XKKM1(J),J=1,N)
      WRITE(4,90) (XKKM1(J),J=1,N)
      YHP(K)=XKKM1(1)
      XHP(K)=XKKM1(2)
      BB=1500+ II*100
 9685 DO 9684 I=1,4
 9686 IF(TIME(K)-BB.LE.0)  GO TO 9997
      BB=BB+6
      IF  (BB-TIME(K).LT.0)  GO TO 9686
 9997 IF(BB-TIME(K).GT.0)GO TO 9699
      YKM6(KKK)=XKK(1)
C
      XKM6(KKK)=XKK(2)
      GO TO 9698
 9699 CC=BB-(1500+II*100)
      WRITE(4,5658)CC,BB
```

```
      WRITE(8,9696)CC,BB
9696  FORMAT(5X,'CC=',F10.4,'    BB=',F10.4,'          HOK')
      IF(CC-24.EQ.0) GO TO 9700
      IF(I-4.LT.0) GO TO 9694
      IF(TIME(K+1)-BB.LT.0) GO TO 9634
9700  BB=BB-24+100
9694  IF(TIME(K+1)-BB.LE.0) GO TO 9634
      IF(BB-TIME(K).EQ.0)GO TO 9698
      DDT=BB-TIME(K)
C
      WRITE(4,3812)BB,DDT,IT, K
      WRITE(8,3812)BB,DDT,IT, K
3812  FORMAT(/,'   BB= ',F10.4,'   DDT= ',F10.4,'    IT= ',I2,
     1 '   K=',I2)
      CALL PHIDEL(DDT,N,L,A,B,PHI,DEL,D1,D2,ND,MD,ID)
      CALL PROD(PHI,XKK,N,N,1,XKKM6,ND,MD,LD)
      WRITE(8,4812)
4812  FORMAT(/,'              XKKM6
     * BB   TIME(K)')
      WRITE(8,90) (XKKM6(J),J=1,N),BB,TIME(K)
      WRITE(4,4812)
      WRITE(4,90) (XKKM6(J),J=1,N),BB,TIME(K)
      YKM6(KKK)=XKKM6(1)
      XKM6(KKK)=XKKM6(2)
9698  XKKK=KKK
      TI=1
      WRITE(4,3969)
3969  FORMAT(/,'  YKM6         XKM6           KKK          BB
     1  I  ')
      WRITE(4,90)YKM6(KKK),XKM6(KKK),XKKK,BB,TI
      KKK=KKK+1
      BB=BB+6
9634  CONTINUE
      WRITE(8,4812)
      WRITE(8,90) (XKKM6(J),J=1,N),BB,TIME(K)
```

```
      WRITE(4,9014)
      WRITE(4,90) (YYKMC(J),J=1,N),ES,TIME(K)
 3222 IT=(TIME(K+1)-1500)*(.01)
      IS=(TIME(K+2)-1500)*(.01)
C
      WRITE(4,9090) IT
 9090 FORMAT(5X,'   IT= ',I4)
      T=24*IT
C
      S=24*IS
      T=TIME(K+1)-IT*100-1500+T
      S=TIME(K+2)-IS*100-1500+S
      DT=S-T
      JK=6
C     WRITE(8,7865)JK
        WRITE(4,555)
        WRITE(8,555)
C       DO 3022 I=1,N
C       WRITE(8,90) (EKKM1(I,J),J=1,N)
C3022   WRITE(4,90) (EKKM1(I,J),J=1,N)
      CALL PHIDEL(DT,N,L,A,B,PHI,DEL,D1,D2,ND,MD,LD)
C     CALL TRANS(DEL,N,1,DELT,ND,MD)
C     CALL PROD(DEL,DELT,N,1,N,Q  ,ND,MD,LD)
C     CALL CONST(W(1,1),Q,N,N,Q,ND,MD)
C  LDLLDLDLDLDLDLLDIDIDLDLDIDLDDLDLDLDLDLDLDIDLDLDLDL
C
C     WRITE(4,544)
C     WRITE(8,544)
C     DO 7134 I=1,N
C     WRITE(4,90)  (Q(I,J),J=1,N)
C7134 WRITE(8,90)  (Q(I,J),J=1,N)
C
C     *** HAS STEP OF FILTER REACHED POINT 25?
      IF((K-25).NE.0)GOTO 2222
C
```

```
     1  ,ICS2(8))
  300   FORMAT(15X,I4,4F9.2)
  310   CONTINUE
C
        RETURN
        END
C
C
C
        SUBROUTINE JUTIME (ITIME,JULHR)
C       ********************* LOGIC ROUTINE **********
C       *** CALCULATES JULIAN TIME FROM YEAR 1900
C       *** IYR=YEAR, IMO=MONTH (MARCH), IDA=DAY,
C       *** IHR=HOUR OF DAY
        IYR=1982
        IMO=3
        IDA=ITIME/100
        IHR=ITIME-IDA*100
        CALL NUMCEN(IYR,IMO,IDA,IHR,JULHR)
        RETURN
C
        END
        SUBROUTINE NUMCEN(YEAR,MO,DA,HR,JULHR)
C       *** CALLED BY SUBROUTINE JUTIME
C       *** CALCULATES JULIAN DAY AND JULIAN HOUR
        INTEGER INID(12),YEAR,DA,HR
        DATA INID/0,31,59,90,120,151,181,212,243,273,304
     1  ,334/
        ID= (YEAR-1900.)*365.25-0.25
        IADD=0
        IF(MOD(YEAR,4) .GT. 0)GOTO 603
        IADD=1
  603   JULDA = INID(MO) + DA + IADD
        IHR = 24. * (ID + JULDA - 1) + HR + 0.5
        IYEAR=YEAR-1900
```

```
C
C *** CALCULATES ERRORS IN POSITION OF KALMAN FILTER
C *** PREDICTIONSAND BEST TRACK VALUES AS A FUNCTION OF
C *** JULIAN TIME AND WRITES THE DATA TO THE FILE
C *** 'KREIGE DATA'.
C
C
      DO 20 I=1,50
      ISTIME=INT(TIME(I))
      CALL JUTIME(ISTIME,JULHR)
      ISJUL(I)=JULHR
      DO 10 J=1,50
      IBEST=INT(BTIME(J))
      CALL JUTIME(IBEST,IBJ)
      IBJUL(J)=IBJ
      IF(IBJUL(J).NE.ISJUL(I))GOTO 10
      EB1(I)=YH(I)-BLAT(J)
      EB2(I)=XH(I)-BLONG(J)
C     EOB1(I)=YH(I)-LAT(I)
C     EOB2(I)=XH(I)-LONG(I)
      EOB1(I)=LAT(I)-BLAT(J)
      EOB2(I)=LONG(I)-BLONG(J)
      JTIME(I)=IBJUL(J)
 10   CONTINUE
 20   CONTINUE
      WRITE(14,290)
      WRITE(6,290)
 290  FORMAT(//,10X,'JTIME',5X,'EB1',6X,'EB2',5X,
     1 'EOB1',5X,'EOB2')
      DO 310 N=1,50
      IF(JTIME(N).EQ.0)GOTO 310
      WRITE(6,300) (JTIME(N),EB1(N),EB2(N),EOB1(N)
     1 ,EOB2(N))
      WRITE(14,300) (JTIME(N),EB1(N),EB2(N),EOB1(N)
```

```
C
C
C
C
      SUBROUTINE MREAD(A,N,M,ND,MD,IREAD)
      DIMENSION A(ND,MD),IREAD(10)
      DO 10 I = 1,N
10     READ(5,20) (A(I,J),J = 1,M)
20    FORMAT(8F10.5)
      RETURN
      END
C
C
C
C
      SUBROUTINE MWRITE(A,N,M,ND,MD,IWRITE)
      DIMENSION A(ND,MD),IWRITE(10)
      DO 10 I = 1,N
      WRITE(4,20)  (I,J,A(I,J),  J = 1,M)
10    WRITE(6,20)  (I,J,A(I,J),  J = 1,M)
20    FORMAT(2(3X,'(',I2,',',I2,')  = ',1PE10.3))
      RETURN
      END
C
C
C
     SUBROUTINE TE[illegible](BLAT,BLONG,TIME,ETIME,XH,YH,LAT
    1 ,LONG)
     DIMENSION TIME(300),BTIME(300),BLAT(300),BLONG(300)
    1 ,YH(300)
     DIMENSION XH(300),EB1(300),EB2(300),IBJUL(300)
    1 ,IEJUL(300),
    * ETIME(300),ECB1(300),ECB2(300)
      REAL*4 LAT(300),LONG(300)
C
```

```
      IF (L.GE.KP) GO TO 20
13    DO 14 J = 1,N
      Z = A(L,J)
      A(L,J) = A(KP,J)
14    A(KP,J) = Z
      DO 15 J = 1,N
      Z = X(L,J)
      X(L,J) = X(KP,J)
15    X(KP,J) = Z
20    IF(ABS(A(L,L)).LE.EP) GO TO 50
30    IF(L.GE.N) GO TO 34
31    LP1 = L+1
      DO 36 K = LP1,N
      IF(A(K,L).EQ.0.) GO TO 36
32    RATIO = A(K,L)/A(L,L)
      DO 33 J = LP1,N
33    A(K,J) = A(K,J)-RATIO*A(L,J)
      DO 35 J = 1,N
35    X(K,J)= X(K,J)-RATIO*X(L,J)
36    CONTINUE
34    CONTINUE
40    DO 43 I = 1,N
      I1 = N+1-I
      DO 43 J = 1,N
      S = 0.
      IF(I1.GE.N) GO TO 43
41    IIP1 = I1 + 1
      DO 42 K = IIP1,N
42    S = S+A(I1,K)*X(K,J)
43    X(I1,J) = (X(I1,J)-S)/A(I1,I1)
      RETURN
50    KER = 2
      RETURN
      END
```

```
      RETURN
   11 IF (Q-1.0) 13,12,13
   12 DO 120 I = 1,N
      DO 120 J = 1,M
  120 C(I,J) = A(I,J)
      RETURN
   13 IF (Q+1.0) 15,14,15
   14 DO 140 I = 1,N
      DO 140 J = 1,M
  140 C(I,J) = -A(I,J)
      RETURN
   15 DO 150 I = 1,N
      DO 150 J = 1,M
  150 C(I,J) = Q*A(I,J)
      RETURN
      END
C
C
C
C
      SUBROUTINE RECIP(N,EP,A,X,KER,M)
      DIMENSION  A(M,M),X(M,M)
      DO 1 I = 1,M
      DO 1 J = 1,M
    1 X(I,J) = 0.
      DO 2 K = 1,N
    2 X(K,K) = 1.
   10 DO  34 L = 1,N
      KP = 0
      Z = 0.
      DO 12 K = L,N
      IF (Z.GE.ABS(A(K,J))) GO TO 12
   11 Z = ABS(A(K,L))
      KP = K
   12 CONTINUE
```

```
C
C
      SUBROUTINE PROD(A,B,N,M,L,C,ND,MD,LD)
      DIMENSION A(ND,MD),B(MD,LD),C(ND,LD)
      DO 1 I = 1,N
      DO 1 J = 1,L
    1 C(I,J) = 0.
      DO 151 I = 1,N
      DO 151 J = 1,L
      DO 151 K = 1,M
  151 C(I,J) =C(I,J) + A(I,K)*B(K,J)
      RETURN
      END
C
C
C
C
      SUBROUTINE TRANS(A,N,M,C,ND,MD)
      DIMENSION  A(ND,MD),C(MD,ND)
      DO 153 I = 1,N
      DO 153 J = 1,M
  153 C(J,I) = A(I,J)
      RETURN
      END
C
C
C
C
      SUBROUTINE CONST(Q,A,N,M,C,ND,MD)
      DIMENSION A(ND,MD),C(ND,MD)
      IF(Q) 11,10,11
   10 DO 100 I = 1,N
      DO 100 J = 1,M
  100 C(I,J) = 0.0
```

```
   33 FORMAT('   I -GH   ')
C     DO 35 I=1,N
C  35 WRITE(6,30) (TEMP(I,J),J=1,N)
C        NOTE HERE PKKM1(I,J) = P(K/K-1) WHERE P(K/K-1) =
      CALL PROD(TEMP,PKKM1,N,N,N,PKK,ND,MD,LD)
        RETURN
        END
C
C
C
C
        SUBROUTINE ADD(A,B,N,M,C,ND,MD)
        DIMENSION A(ND,MD),B(ND,MD),C(ND,MD)
C       DO   2 I = 1,N
C       DO   2 J = 1,M
C   2   C(I,J) =0.
        DO 152 I = 1,N
        DO 152 J = 1,M
  152   C(I,J) = A(I,J) + B(I,J)
      RETURN
        END
C
C
C
C
        SUBROUTINE SUB(A,B,N,M,C,ND,MD)
        DIMENSION A(ND,MD),B(ND,MD),C(ND,MD)
        DO 152 I = 1,N
        DO 152 J = 1,M
  152   C(I,J) = A(I,J) - B(I,J)
        RETURN
        END
C
C
```

```
C   24 WRITE(8,90) (TEMP1(I,J),J=1,M)
       CALL ADD(TEMP1,R,M,M,TEMP1,ND,MD)
       CALL RECIP(M,0.0000001,TEMP1,TEMP2,KER,MD)
C      TEMP2(1,1)=TEMP3(2,2)/DET
C      TEMP2(2,1)=-TEMP3(2,1)/DET
C      TEMP2(1,2)=-TEMP3(2,1)/DET
C      TEMP2(2,1)=-TEMP3(1,1)/DET
C      WRITE(8,31)
C      DO 27 I=1,M
C   27 WRITE(8,90) (TEMP2(I,J),J=1,M)
    31 FORMAT('     (HPH +R)-1')
       IF (KER-2) 101,110,101
   110 WRITE(8,111)
   111 FORMAT (5HKER=2)
   101 CALL PROD(TEMP,TEMP2,N,M,M,G,ND,MD,LD)
C  NOTE HERE PKK(I,J) = P(K/K) WHERE P(K/K) =
C   (I-G(K)*H)*P(K/K-1)
       CALL PROD(G,H,N,M,N,TEMP,ND,MD,LD)
C      WRITE(8,30)
C   30 FORMAT('     GH     ')
C      DO 25 I=1,N
C   25 WRITE(8,90) (TEMP(I,J),J=1,N)
       DO 108 I = 1,N
       DO 108 J = 1,N
   108 TEMP(I,J) = -TEMP(I,J)
C      WRITE(8,37)
    37 FORMAT('     HI     ')
C      DO 45 I=1,N
C   45 WRITE(8,90) (  HI(I,J),J=1,N)

       CALL ADD(HI,TEMP,N,N,TEMP,ND,MD)
C      WRITE(8,33)
```

```
      1    ,MD,LD,K)
       DIMENSION PKK(12,12),Q(12,12),H(12,12),G(12,12)
      1 ,R(12,12),
      1   HI(12,12),HT(12,12),TEMP(12,12),TEMP1(12,12),
      1 TEMP2(12,12),
      &PHI(12,12),PHIT(12,12),PKKM1(12,12)
C      G(K)  = P(K/K-1)*HI*(H*P(K/K-1)*HI + R)
C        PHI*P(K-1/K-1)*PHIT  + Q
       IF(K.EQ.1) GO TO 8889
         CALL TRANS(PHI,N,N,PHIT,ND,MD)
         CALL PROD(PKK,PHIT,N,N,N,TEMP,ND,MD,LD)
         CALL PROD(PHI,TEMP,N,N,N,TEMP1,ND,MD,LD)
         CALL ADD(TEMP1,Q,N,N,PKKM1,ND,MD)
         WRITE(8,555)
  555    FORMAT(/'     MATRIX PKKM1 ')
      DO 3022 I=1,N
      WRITE(8,90) (PKKM1(I,J),J=1,N)
 3022 WRITE(4,90) (PKKM1(I,J),J=1,N)
 8889 CONTINUE
      CALL TRANS(H,M,N,HI,ND,MD)
C     WRITE(8,39)
   39 FORMAT('        H     ')
C     DO 22 I=1,M
C  22 WRITE(8,90)(H(I,J),J=1,N)
   90 FORMAT(1P7E11.4)
C     WRITE(8,36)
   36 FORMAT('        HI    ')
C     DO 23 I=1,N
C  23 WRITE(8,90)(HI(I,J),J=1,M)
      CALL PROD(PKKM1,HI,N,N,M,TEMP,ND,MD,LD)
      CALL PROD(H,TEMP,M,N,M,TEMP1,ND,MD,LD)
C     WRITE(8,38)
   38 FORMAT('  H P HI ')
C     DO 24 I=1,M
```

```
      C(IR,IC) = A(IR,IC)
      TEIL(IR,IC) = T/2.D0*PHI(IR,IC)
10    TERM(IR,IC) = T*PHI(IR,IC)
50    DO 11 IR = 1,N
      DO 11 IC = 1,N
      COR(IR,IC) = T/F*C(IR,IC)
      PHI(IR,IC) = PHI(IR,IC)+COR(IR,IC)
      TEIL(IR,IC) = TEIL(IR,IC)+T/((F+1.   )*(F+2.   ))
     1  *COR(IR,IC)
11    TERM(IR,IC) = TERM(IR,IC)+T/(F+1.   )*COR(IR,IC)
      DO 12 IR = 1,N
      DO 12 IC = 1,N
      C(IR,IC) = 0.
      DO 12 K = 1,N
12    C(IR,IC) = C(IR,IC)+A(IR,K)*COR(K,IC)
      F = F+1.
      DO 13 IR = 1,N
      DO 13 IC = 1,N
      IF(ABS(COR(IR,IC)).GT.TEST*ABS(PHI(IR,IC)))
     1  GO TO 50
13    CONTINUE
      CALL PROD(TERM,B,N,N,M,DEL,ND,MD,ID)
      CALL PROD(TEIL,B,N,N,M,D2,ND,MD,ID)
      DO 14 IR = 1,N
      DO 14 IC = 1,M
14    D1(IR,IC) = DEL(IR,IC)-D2(IR,IC)
      RETURN
      END
C THIS SUBROUTINE COMPUTES THE OPTIMUM GAIN MATRIX AND THE
C COVARIANCE
C
C
C
C
      SUBROUTINE GAIN(PKN,PKK1,Q,R,PHI,H,N,M,G,NI,ND
```

```
      WRITE(10,2429)
 2429 FORMAT ('INDEX',3X,'TIME',4X,'G11',6X,'G31',6X,'YH'
     *,6X, 'YHP',5X,'LONG',5X,'LAT',7X,'XH')
C
C                   VALUES TO BE PLOTTED
      WRITE (10,2430) (I,TIME(I),G11(I),G31(I),YH(I),YHP(I),
     *LONG(I),LAT(I),XH(I), I = 1,NN)
C
 2430 FORMAT (I4,8F9.2)
C
C *********************************************************
      WRITE(8,1016) K
C     WRITE(9,1016) K
      DO 65 J=1,N
      WRITE(8,90) (G(J,I),I=1,M)
   65 CONTINUE
      WRITE(8,16)
   16 FORMAT(' ',/,3X,'COV. MAT. OF PREDICTED ESTIMATE' )
      DO 15 J=1,N
      WRITE(8,90) (PKKM1(I,J),I=1,N)
   15 CONTINUE
  999 STOP
        END
CWWWWMWWMWWWWWWMWWWMWMWWMWMWWWWMWWMMWWMWWMWMWMWWMWWWWWWWWWWWMY
C
        SUBROUTINE PHIDEL(T,N,M,A,B,PHI,DEL,D1,D2,ND,MD,LD)
        DIMENSION A(12,12),B(12,12),PHI(12,12),DEL(12,12),
     *  TERM(12,12),
     *  COR(12,12),C(12,12),D1(12,12),D2(12,12),TEIL(12,12)
        TEST = 1.E-7
        F=1.
        DO 10 IR = 1,N
        DO 10 IC = 1,N
        PHI(IR,IC) = 0.
        PHI(IR,IR) = 1.
```

```
  327 CONTINUE
      WRITE(4,333) (BTIME(I),BLAT(I),BLONG(I),BY(I),BX(I)
     1   ,I=1,N2)
      WRITE(8,9898) (G11(K),K=1,10)
      WRITE(8,9898) (TIME(K),K=1,10)
 9898 FORMAT(F10.4)
      WRITE(8,410)
  410 FORMAT('1')
C     WRITE(4,7777)
C
C     CALL PLOTP(TIME,G11,NN,1)
C     CALL PLOTP(TIME,G31,NN,3)
      WRITE(8,410)
C     WRITE(4,7777)
C     CALL PLOTP(TIME,YH,NN,1)
C     CALL PLOTP(TIME,YHP,NN,3)
      LL=NN+2
      LONG(LL-1)=100
      LONG(LL)= 160
      LAT(LL-1)=0
      LAT(LL)=60
      WRITE(4,7777)
C     CALL PLOTT(LONG,LAT,LL,1)
C     CALL PLOTT(XH,YH,NN,3)
C     CALL PLOTP(LONG,LAT,LL,1)
C     CALL PLOTP(XH,YH,NN,3)
C
C ***** WRITE VALUES INTO PLOT FILE FOR DISSPLA **********
C
C               NUMBER OF VALUES
      WRITE(10,2428) NN,W(1,1),R(1,1)
      WRITE( 4,2428) NN,W(1,1),R(1,1)
 2428 FORMAT(I4,3X,'W(1,1)=',F8.3,3X,'R(1,1)=',F8.2)
C
C               COLUMN HEADINGS
```

```
2223  FORMAT(4I10)
C
*************************************************************
C      WRITE(4,7777)
       WRITE(6,393) (TIME(I),LAT(I),LONG(I),YH(I),XH(I),I=1
      1   ,NZ)
       WRITE(6,393)  (BTIME(I),BLAT(I),BLONG(I),YKM6(I)
      1  ,XKM6(I),I=1,NZ)
  393 FORMAT (1X,5F 8.2)
C      WRITE(4,7777)
       WRITE(4,393)  (TIME(I),LAT(I),LONG(I),YH(I),XH(I)
      1     ,I=1,NZ)
       WRITE(4,393)  (BTIME(I),BLAT(I),BLONG(I),YKM6(I)
      1    ,XKM6(I),I=1,NZ)
C
       INTIME=INT(TIME(K))
       FLTIME=FLOAT(INTIME)
       IF(FLTIME.NE.TIME(K))GOTO 399
       CALL TERM(BLAT,BLONG,TIME,BTIME,XH,YH,LAT,LONG)
  399  CONTINUE
C      *** WRITE EOB1 AND EOB2 TO FILE 'EOB DATA'
       WRITE(6,329)
  329  FORMAT(//,5X,'   JULIAN HR          EOB1
      1  EOB2')
       WRITE(6,328)(KJULHR(IEOB),EOB1(IEOB),EOB2(IEOB)
      1  ,IEOB=1,NZ)
       WRITE(16,328)(KJULHR(IEOB),EOB1(IEOB),EOB2(IEOB)
      1  ,IEOB=1,NZ)
  328  FORMAT(I15,2F15.2)
C
       DO 327 I=1,NN
       EY(I)=YKM6(I)-BLAT(I)
       EX(I)=XKM6(I)-BLONG(I)
       TIME(I)=FLOAT(I)
```

```
C
C      *** ROUTINE TO PLACE ELLIPSE DATA IN FILE
C
       THE1=.50*ATAN(2*PKKM1(1,2)/(PKKM1(1,1)-PKKM1(2,2)))
       SIG2X=(PKKM1(1,1)+PKKM1(2,2))/2.+PKKM1(1,2)
      1  /SIN(2.*THE1)
       SIG2Y=(PKKM1(1,1)+PKKM1(2,2))/2.-PKKM1(1,2)
      1  /SIN(2.*THE1)
       SX=((SIG2X)**.5)*5
       SY=((SIG2Y)**.5)*5
       PI=3.14159265/12
       CT=COS(THE1)
       ST=SIN(THE1)
       DO 1981 IELLIP=1,25
       XI=IELLIP
       XP(IELLIP)=SX*COS(PI*XI)*CT-SY*SIN(PI*XI)*ST+XH(25)
       YP(IELLIP)=SX*COS(PI*XI)*ST+SY*SIN(PI*XI)*CT+YH(25)
1981   WRITE(15,1982) XP(IELLIP),YP(IELLIP)
1982   FORMAT(2F15.7)
C      *** END OF ELLIPSE CALCULATION
C
C
 2222  CONTINUE
C
C      **** WRITE IKM STUFF
       WRITE(8,2224)
2224   FORMAT(//,' IKM72(JJ), IKM48(JJ), IKM24(JJ)
      1  , IKM6(JJ) ')
       WRITE(6,2223) (IKM72(JJ),IKM48(JJ),IKM24(JJ),IKM6(JJ)
      1  ,JJ=1,NN)
       WRITE(8,2225)
2225   FORMAT(//,' IKP72(MM), IKP48(MM), IKP24(MM),
      1  IKP6(MM) ')
       WRITE(6,2223) (IKP72(MM),IKP48(MM),IKP24(MM),IKP6(MM)
      1  ,MM=1,NN)
```

```
JULST=IND
RETURN
END
```

BIBLIOGRAPHY

Kirk, D.E., "Optimal Estimation: An Introduction to the Theory and Applications", Mimeographed Notes for course EE4413, Naval Postgraduate School, Monterey, CA, Spring 1984.

Maybeck, P.S., Stochastic Models, Estimation, and Control, Vol. 1, 2, New York: Academic Press, 1982.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
2.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
3.	Hellenic Navy General Staff Geniko Epiteleio Nautikou Stratopedo Papagou Athens, Greece	1
4.	Professor Harold A. Titus, 62Ti Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943	2
5.	Professor Alex Gerba, Jr., Code 62Gz Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943	1
6.	Department Chairman, Code 62Rr Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943	1
7.	LT Pantos Theodore, H.N. Vassileos Pavlou 30, Palaia Peudeli Athens, Greece	1
8.	LT V. Martzoucos, H.N. Leoforos Galatsiou 115 Athens, Greece	1
9.	LT. V. Tsafaras, H.N. Dagli 27, Kifissia Athens, Greece	3